



Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

**Optimización de estructuras de
acero utilizando diseño
paramétrico y algoritmos genéticos
en un entorno BIM**

Trabajo realizado por:

Marco Antonio Cisneros Cantú

Dirigido por:

Dr. Rolando Antonio Chacón Flores

Màster en:

Ingeniería Estructural y de la Construcción

Barcelona,

Departamento de Ingeniería Civil y Ambiental

TRABAJO FINAL DE MÀSTER

Agradecimientos

- En primer lugar agradezco a mis padres por todo el esfuerzo y apoyo que siempre me han dado durante todos mis años de estudios y haberlos hecho posibles.
- A mi novia y su familia por su apoyo y comprensión durante mis estudios de máster.
- Agradezco a mi tutor el Dr. Rolando Antonio Chacón Flores por todo su apoyo durante este TFM, el conocimiento que me brindó, consejos, tiempo y experiencia.
- A todos los profesores que tuve durante mis máster en la UPC que me proporcionaron sus conocimientos y experiencia.

Contenido

AGRADECIMIENTOS.....	2
ÍNDICE DE FIGURAS.....	5
ÍNDICE DE TABLAS.....	10
RESUMEN.....	11
ABSTRACT.....	11
1 INTRODUCCIÓN	12
1.1 OBJETIVO GENERAL.....	12
1.1.1 <i>Objetivos Específicos</i>	12
1.2 ALCANCE	12
1.3 METODOLOGÍA	13
2 ESTADO DEL ARTE.....	16
2.1 DISEÑO PARAMÉTRICO	16
2.2 OPTIMIZACIÓN ESTRUCTURAL.....	22
2.2.1 <i>Optimización metaheurística</i>	24
2.2.2 <i>Algoritmos evolutivos genéticos</i>	24
2.2.3 <i>Investigaciones recientes sobre optimización estructural</i>	25
2.3 INVESTIGACIONES RECIENTES SOBRE OPTIMIZACIÓN ESTRUCTURAS Y DISEÑOS PARAMÉTRICOS	29
3 HERRAMIENTAS	35
3.1 RHINOCEROS 3D	35
3.2 GRASSHOPPER.....	37
3.3 KARAMBA 3D	39
3.4 GALÁPAGOS.....	42
3.5 OCTOPUS	46
4 CASOS DE ESTUDIO.....	49
4.1 PRIMER CASO DE ESTUDIO.....	51
4.1.1 <i>Criterio para mayoración de cargas</i>	52
4.1.2 <i>Modelación paramétrica</i>	52
4.1.3 <i>Simulación estructural</i>	53
4.1.4 <i>Definición de los apoyos</i>	54

4.1.5	<i>Definición de las cargas</i>	54
4.1.6	<i>Definición de materiales</i>	55
4.1.7	<i>Ensamble del modelo</i>	56
4.1.8	<i>Análisis estructural y su resultado</i>	56
4.1.9	<i>Optimización del perfil</i>	59
4.2	SEGUNDO CASO DE ESTUDIO	65
4.2.1	<i>Criterio para mayoración de cargas</i>	65
4.2.2	<i>Modelación paramétrica</i>	65
4.2.3	<i>Simulación estructural</i>	67
4.2.4	<i>Definición de los apoyos</i>	68
4.2.5	<i>Definición de las cargas</i>	69
4.2.6	<i>Definición de materiales</i>	70
4.2.7	<i>Ensamblaje del modelo</i>	72
4.2.8	<i>Optimización topológica</i>	73
4.2.9	<i>Análisis estructural y su resultado</i>	74
4.2.10	<i>Optimización estructural</i>	76
4.2.11	<i>Optimización a partir de Galápagos</i>	77
4.2.12	<i>Optimización a partir de Octopus y Galápagos</i>	86
4.3	TERCER CASO DE ESTUDIO	91
4.3.1	<i>Criterio para mayoración de cargas</i>	92
4.3.2	<i>Modelación paramétrica</i>	92
4.4	SIMULACIÓN ESTRUCTURAL	94
4.4.1	<i>Optimización estructural</i>	101
4.4.2	<i>Optimización con Galápagos</i>	101
4.4.3	<i>Optimización con Octopus</i>	104
5	DISCUSIÓN DE RESULTADOS	106
6	CONCLUSIONES Y RECOMENDACIONES	108
7	BIBLIOGRAFÍA	110
	ANEXOS	113

Índice de Figuras

Figura 1. Primer Caso de Estudio, Viga Simplemente Apoyada.....	14
Figura 2. Segundo Caso de Estudio, Celosía Empotrada.....	14
Figura 3. Tercer Caso de Estudio, Viga Simplemente Apoyada	15
Figura 4. Curva de Paulson. Citado de (Paulson, 1976)	17
Figura 5. Pabellón Paramétrico. Citado de (Rodríguez, 2015)	17
Figura 6. Distintas Soluciones Diseño Paramétrico. Citado de (Ruiz, 2015)	18
Figura 7. Pasos Método CIBIM. Citado de (Lee & Ha, 2013)	19
Figura 8. Resultado Facha Paramétrica Ante Distinta Luz Solar. Citado de (Kensek, 2014) .	20
Figura 9. Modelos Generados con Distintos Parámetros. Citado de (Pan, Turrin, Louter, Sariyildiz, & Sun, 2019)	21
Figura 10. Programas Actuales para Diseño Paramétrico	22
Figura 11. Ejemplo de Optimización de Tamaño	23
Figura 12. Ejemplo de Optimización de la Geometría	23
Figura 13. Ejemplo de Optimización Topológica.....	23
Figura 14. Resultados Obtenidos Por Diferentes Autores y Algoritmos. Citado de (Sánchez, 2012)	25
Figura 15. Comparación de Resultados Método SESO y ESO. Citado de (Simonetti, Almeida, & Neto, A smooth evolutionary structural optimization procedure applied to plane stress problem, 2014).....	26
Figura 16. Proceso Propuesto de Impresión 3D de Concreto. Citado de (Costanzi, y otros, 2018)	27
Figura 17. Viga en Voladizo Utilizada en Optimización Topológica. Citada de (Simonetti H. L., Almeida, Leitao, & Neves, 2019).....	28
Figura 18. Viga Simplemente Apoyada Utilizada en Optimización Topológica. Citada de (Simonetti H. L., Almeida, Leitao, & Neves, 2019).....	28
Figura 19. Modelo Paramétrico. Citado de (Taleb & Musleh, 2015)	30
Figura 20. Parámetros de las Edificaciones. Citado de (Taleb & Musleh, 2015).....	31
Figura 21. Posibles Patrones de Diagrama Durante Optimización. Citado de (Tomei, Imbimbo, & Mele, 2018)	31
Figura 22. Comparación de Derivas entre los Patrones. Citado de (Tomei, Imbimbo, & Mele, 2018)	32

Figura 23. Software y su Proceso de Comunicación. Citado de (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018).....	33
Figura 24. Parámetros Externos de Estructura Exterior Predefinidas. Citado de (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018)	34
Figura 25. Logotipo de Rhinoceros 3D. Citado de (McNeel & Associates, 2019)	35
Figura 26. Ejemplo de uso de NURBS. Citado de (McNeel, 2019)	36
Figura 27. Estructura Generada en Rhinoceros 3D. Citado de (McNeel & Associates, 2019).....	36
Figura 28. Logotipo de Grasshopper. Citado (McNeel & Associates, 2019)	37
Figura 29. Interfaz de Grasshopper. Citado de (Gil & Parsons, 2014).....	38
Figura 30. Programación Visual en Grasshopper y su Modelo Visual en Rhino, Citado de (Gil & Parsons, 2014).....	39
Figura 31. Proceso de Modelar una Curva a Través de Grasshopper en Distintos Métodos. Citado de (Gil & Parsons, 2014).....	39
Figura 32. Logotipo de Karamba 3D. Citado de (Preisinger, Karamba 3D Parametric Enginerring, 2019)	40
Figura 33. Karamba 3D dentro de Grasshopper. Citado de (Preisinger, Karamba 3D Parametric Enginerring, 2019).....	40
Figura 34. Viga en Cantilever Modelada Utilizando Grasshopper y Karamba 3D. Citado por (Preisinger, Karamba 3D Parametric Enginerring, 2019).....	41
Figura 35. Logotipo de Galápagos. Citado de (Rutten, 2010)	42
Figura 36. Modelo de Ejemplo Galápagos. Citado de (Rutten, 2010).....	43
Figura 37. Población de Individuos del Ejemplo. Citado de (Rutten, 2010)	44
Figura 38. Limpieza de Genes Menos Aptos. Citado de (Rutten, 2010)	44
Figura 39. Interfaz de Galápagos Pestaña de Opciones	45
Figura 40. Interfaz de Galápagos Pestaña de Solvers	46
Figura 41. Logotipo de Octopus. Citado de (Vierlinger, 2012).....	46
Figura 42. Interfaz de Octopus. Citado de (Vierlinger, 2012)	47
Figura 43. Viga Simplemente Apoyada (Primer Caso de Estudio)	51
Figura 44. Coeficientes de mayoración. Citado de (EN 1990, 2002)	52
Figura 45. Definición Geométrica de la Viga Programada.....	53
Figura 46. Modelo Visual de la Geometría de la Viga	53
Figura 47. Definición del Elemento en Karamba 3D	53
Figura 48. Definición de los Nodos en el Elemento	54

Figura 49. Definición de Apoyos.....	54
Figura 50. Deifnición de Cargas en Karamba 3D	55
Figura 51. Catálogo de Perfiles de Acero	56
Figura 52. Ensamble del Modelo	56
Figura 53. Análisis Estructural del Elemento	57
Figura 54. Resultados del Modelo	57
Figura 55. Momento Actuante en Estado Límite de Servicio.....	58
Figura 56. Momento Actuante es Estado Límite Último	58
Figura 57. Complemento de Optimización Karamba 3D	59
Figura 58. Anejo B Método 2 para Obtener Factores kij. Citado de (EN 1993-1-1, 2005).....	60
Figura 59. Valores Límite Recomendados para Flechas Verticales. Citado de (Gulvanessian & Holicky, 1996)	60
Figura 60. Resultado de Optimización.....	61
Figura 61. Componente de Utilización de Elementos	62
Figura 62. Complemento de Catálogo con Perfiles IPE	63
Figura 63. Perfil IPE 400, Obtenido con Nuevo Catálogo	63
Figura 64. Programación Gráfica en Segundo Caso de Estudio.....	65
Figura 65. Definición Geométrica de la Celosía Programada	66
Figura 66. Modelo Visual de la Geometría de la Celosía	66
Figura 67. Definición de Geometría como Elemento en Karamba 3D.....	67
Figura 68. Clasificación de los Elementos de acuerdo con ID's	67
Figura 69. Definición de Apoyos Caso 2.....	68
Figura 70. Componente para Definir Articulaciones	68
Figura 71. Definición de Carga y Casos de Carga en Karamba 3D	69
Figura 72. Catálogo de Perfiles y Definición de Materiales	70
Figura 73. Componente de Selección por Rango de Perfiles.....	71
Figura 74. Componente de Selección de Material	71
Figura 75. Componentes Deslizantes para Seleccionar Perfil	71
Figura 76. Selector de Perfiles	72
Figura 77. Ensamble del Modelo	72
Figura 78. Optimización Topológica Celosía	73
Figura 79. Sucesión de Aproximaciones en Optimización Topológica.....	73
Figura 80. Componente de Análisis Estructural y Desplazamientos	74

Figura 81. Resultados del Modelo	75
Figura 82. Componente de Esfuerzos Resultantes en la Estructura.....	75
Figura 83. Numeración de los Elementos en la Celosía	76
Figura 84. Componente Galápagos.....	77
Figura 85. Componente de Función para Optimizar.....	77
Figura 86. "Genome" del Componente Galápagos	78
Figura 87. Componente en Código para Limitar Canto.....	79
Figura 88. Venta Para Programar en Código del Componente de Visual Basic.....	79
Figura 89. Componente en Código para Limitar Flecha.....	80
Figura 90. Componente de Utilización	81
Figura 91. Componente en Código para Verificar Ratios.....	82
Figura 92. Componente en Código para Convertir a Mil	83
Figura 93. Datos Objetivo de Galápagos	83
Figura 94. Componente Galápagos Durante Simulación.....	84
Figura 95. Estructura Obtenida con Galápagos	84
Figura 96. Ratios de Utilización Obtenidos por Galápagos.....	86
Figura 97. Optimización Geométrica Octopus	87
Figura 98. Función para Optimizar Ratios de Utilización en Galápagos	87
Figura 99. Modelo Primera Iteración con Octopus y Galápagos.....	88
Figura 100. Modelo Segunda y Tercera Iteración con Octopus y Galápagos	88
Figura 101. Numeración de Elementos Segunda y Tercera Iteración con Octopus y Galápagos	89
Figura 102. Programación Gráfica en Tercer Caso de Estudio	91
Figura 103. Modelación Geométrica de Viga Armada.....	92
Figura 104. Vista Frontal Viga Armada	92
Figura 105. Sección Transversal Viga Armada	93
Figura 106. Vista en Perspectiva Viga Armada.....	93
Figura 107. Componente en Código para Analizar Viga Armada.....	94
Figura 108. Relaciones Máximas Ancho/Espesor para Miembros a Compresión (Determinar Clase del Alma). Citado de (EN 1993-1-1, 2005)	95
Figura 109. Relaciones Máximas Ancho/Espesor para Miembros a Compresión (Determinar Clase de Patín). Citado de (EN 1993-1-1, 2005)	96

Figura 110. Relación entre Factores de Pandeo y Esfuerzo en Base a Condiciones de Contorno. Citado de (EN 1993-1-5, 2006)	97
Figura 111. Contribución del alma χ_w a Resistencia a Pandeo por Cortante	98
Figura 112. Valores Recomendados para Factores de Imperfección para Curvas de Pandeo Lateral. Citado de (EN 1993-1-1, 2005)	99
Figura 113. Valores Recomendados para Curvas de Pandeo Lateral para Secciones Transversales. Citado de (EN 1993-1-1, 2005)	99
Figura 114. Factores de Corrección K_c . Citado de (EN 1993-1-1, 2005)	100
Figura 115. Parámetros para Modificar por Galápagos en Optimización	102
Figura 116. Función de Optimización Tercer Caso de Estudio	102
Figura 117. Resultados Optimización Galápagos en Tercer Caso de Estudio.....	103
Figura 118. Resultados Optimización Octopus en Tercer Caso de Estudio	104

Índice de Tablas

Tabla 1. Magnitud y Posición de Cargas	55
Tabla 2. Caso de Carga y Factores de Amplificación	55
Tabla 3. Resumen de Resultados	59
Tabla 4. Resumen de Parámetros Utilizados en Verificaciones de Perfil HEB 220.....	62
Tabla 5. Resumen de Parámetros Utilizados en Verificaciones de Perfil IPE 400.....	64
Tabla 6. Magnitud y Posición de Carga.....	69
Tabla 7. Casos de Carga y Factores de Amplificación	69
Tabla 8. Resumen de Resultados	76
Tabla 9. Perfiles Obtenidos con Galápagos	85
Tabla 10. Parámetros se la Celosía Obtenidos con Galápagos	85
Tabla 11. Perfiles Seleccionados para Celosía en Primera Iteración.....	88
Tabla 12. Parámetros de Celosía Obtenidos en Primera Iteración.....	88
Tabla 13. Perfiles Seleccionados para Celosía en Segunda y Tercera Iteración	89
Tabla 14. Parámetros de Celosía Obtenidos en Segunda y Tercera Iteración	89
Tabla 15. Comparación de Resultados entre Galápagos y Octopus en Conjunto con Galápagos.....	90
Tabla 16. Resumen de Resultado Optimización Galápagos Tercer Caso de Estudio.....	104
Tabla 17. Resumen de Resultado Optimización Octopus Tercer Caso de Estudio	105
Tabla 18. Comparación de Resultados entre Galápagos y Octopus	105

Resumen

En el sector de la construcción las empresas cada vez requieren estructuras diseñadas de la manera más óptima posible, para reducir costes y hacerlas más sostenibles. Optimizar las estructuras se ha convertido en una necesidad indiscutible hoy en día. Con el paso del tiempo las técnicas de optimización han evolucionado y se han desarrollado algoritmos más potentes, este hecho exige indirectamente una evolución en la manera en que se modelan los proyectos para brindar mejores resultados y más rápidos al momento optimizar las estructuras, estos modelos son llamados modelos paramétricos, los cuales permiten al usuario realizar grandes modificaciones de manera rápida sin tener que redimensionar el proyecto ante cambios en su configuración. En este TFM se desarrolló una metodología de cálculo para optimizar tres casos de estudio mediante algoritmos a partir de modelos paramétricos en un entorno BIM. Los casos analizados incluyen una viga laminada, una celosía y una viga armada. La metodología aplicada en los tres casos de estudio demostró tener un buen funcionamiento, ya que los modelos paramétricos trabajando en conjunto con los algoritmos de optimización genética generaron resultados que se ajustaron de manera adecuada a las verificaciones de resistencia y estabilidad del Eurocódigo 3. Los ratios de utilización y las deformaciones (ULS y SLS) se ajustaron a los límites establecidos.

Abstract

In the construction sector, companies increasingly require structures designed as optimally as possible, both to reduce costs and to add sustainability. Optimizing structures has become an indisputable necessity today. Over time, optimization techniques have evolved and more powerful algorithms have been developed, this fact indirectly demands an evolution in the way in which projects are modeled to provide better results and faster when optimizing structures, these models are called parametric models, which allow the user to make large modifications quickly without having to start a model from scratch. In this thesis a calculation methodology was developed to optimize three case studies using algorithms based on parametric models in a BIM environment. The cases analyzed include a laminated beam, a lattice and an armed beam. The methodology applied in the three case studies proved to have a good functioning, since the parametric models working together with the genetic optimization algorithms generated results that were adjusted in an adequate way to the resistance and stability verifications of Eurocode 3 and the ratios of use and deflection were adjusted to the limits that were established.

1 Introducción

1.1 Objetivo General

Desarrollar metodología de cálculo basada en diseño paramétrico y algoritmos genéticos en un entorno BIM que permita optimizar una estructura.

1.1.1 Objetivos Específicos

- Modelar paramétricamente elementos estructurales de acero, utilizando programación gráfica de algoritmos mediante el software Rhinoceros 3D versión 6.17 y su complemento Grasshopper.
- Desarrollar un análisis estructural de elementos metálicos adaptado a la variación paramétrica permitida en los modelos utilizando Karamba 3D versión 1.3.2, complemento de Grasshopper .
- Optimizar los elementos estructurales de acero generados en los modelos utilizando el complemento de Grasshopper llamado Galapagos y Octopus, considerando como optimo el elemento menos pesado, una utilización lo más cercana posible al 100 por ciento y que se encuentre lo más cercano posible a la flecha mínima permisible, para obtener estos resultados se utilizan las verificaciones del Eurocódigo 3 (EN 1993-1-1, 2005).

1.2 Alcance

En esta tesis se analizan tres casos de estudio para verificar el comportamiento de la optimización de dichos casos a través de algoritmos evolutivos genéticos aplicado por los complementos Galápagos y Octopus. Se analizan dos casos de vigas simplemente apoyadas y un caso de una celosía empotrada, todos con carga permanente. Los casos analizados se verifican y optimizan en cuanto a resistencia y estabilidad a partir del Eurocdico 3, también se verifica que cumplan con una flecha máxima admisible.

1.3 Metodología

La primera parte de la tesis consistió en obtener conocimiento fundamental sobre Grasshopper para aprender a realizar programación gráfica, una vez que se logró este objetivo se procedió a estudiar el complemento de Grasshopper llamado Karamba 3D que sirve para modelar y analizar elementos estructurales. Con el conocimiento suficiente del software a utilizar se realizan tres casos de estudio para aplicar distintos métodos de optimización. Cabe destacar que el entorno de programación visual Grasshopper incluye una total interoperabilidad con entornos BIM (por ejemplo, Tekla) lo que permite de manera directa trabajar en el mismo entorno.

Se compararon los resultados de optimización obtenidos a través de Galápagos y Octopus, ambos complementos trabajan a partir de algoritmos evolutivos pero el primero trabaja utilizando objetivos simples para obtener resultados y el segundo utiliza multi objetivos para realizar el mismo procedimiento, es por eso por lo que resulta interesante realizar una comparación entre los resultados de ambos complementos. Para la optimización se tomaron en cuenta los siguientes criterios partiendo del Eurocódigo 3:

- Flecha máxima permisible en estado límite de servicio.
- Capacidad seccional de los elementos en estado límite último.
- Estabilidad de los elementos en estado límite último.
- Peso mínimo de los elementos.
- Ratio de utilización máximo.

Los casos de estudio que se analizaron son los siguientes:

- El primer caso analizado mostrado en la Figura 1 es una viga simplemente apoyada con carga permanente aplicada como una carga puntual al centro de la viga y un axil al extremo, verificando su comportamiento y optimizándolo con un complemento interno de Karamba 3D que obtiene los posibles elementos a partir de catálogos que incluyen los posibles perfiles a utilizar y realiza verificaciones a partir del Eurocódigo 3 para seleccionar el perfil óptimo.

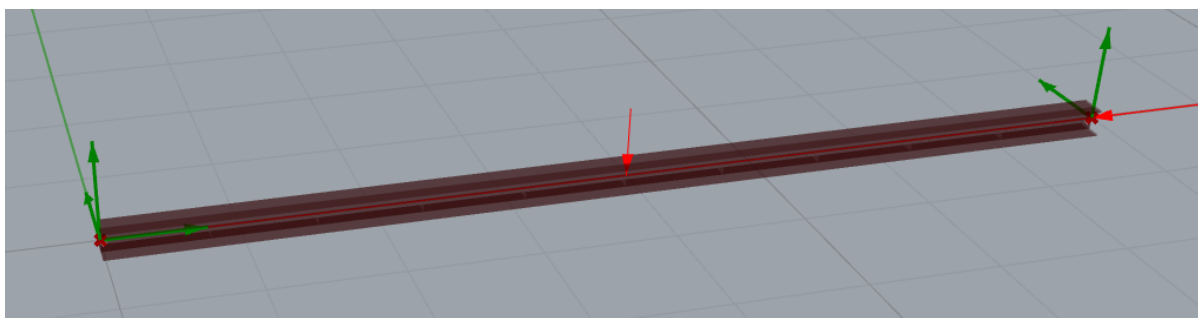


Figura 1. Primer Caso de Estudio, Viga Simplemente Apoyada

- El segundo caso de análisis mostrado en la Figura 2 es una celosía empotrada con carga permanente aplicada en de manera puntual al centro de la estructura, la optimización es más compleja, se encuentran nuevas variantes como el canto de la celosía, el número de divisiones y los perfiles a utilizar. El complemento optimizador de Karamba 3D ya no es suficiente por lo que ahora es necesario utilizar los complementos de Grasshopper que son Galápagos y Octopus que permiten buscar la mejor opción a partir de algoritmos evolutivos.

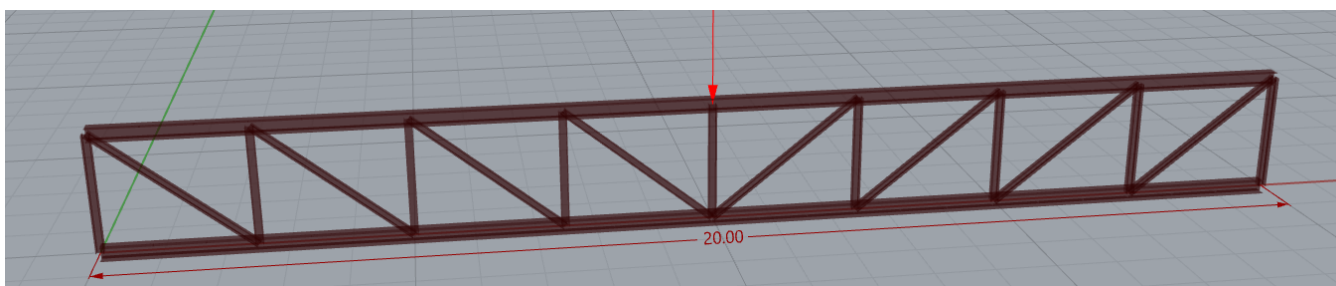


Figura 2. Segundo Caso de Estudio, Celosía Empotrada

- El tercer caso de análisis en la Figura 3 es una viga simplemente apoyada con carga permanente en forma de carga distribuida, a diferencia de la viga en el primer caso de estudio, esta viga es una viga armada en donde el perfil ya no puede ser obtenido a través de catálogos, al tener este impedimento no se puede trabajar con Karamba 3D, ya que solamente utiliza perfiles existentes. Para este caso se requiere de programación haciendo una combinación entre complementos de Grasshopper que permiten programar lo necesario para realizar las verificaciones estructurales utilizando Python y Visual Basic y haciendo uso de Galápagos y Octopus nuevamente para optimizar el elemento.

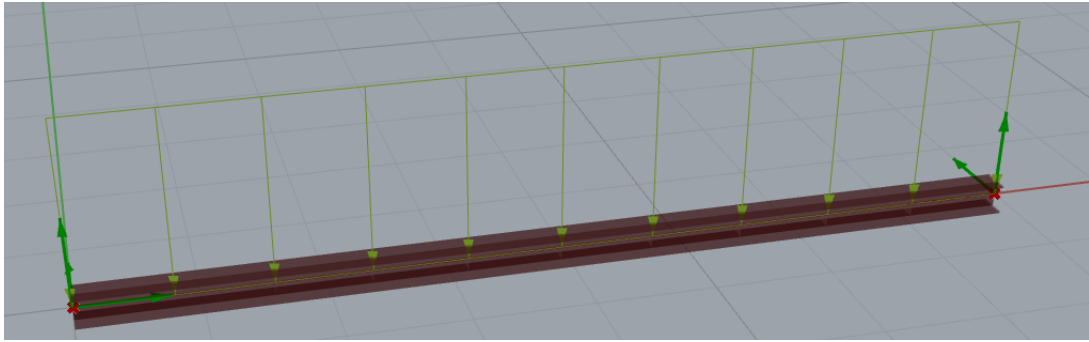


Figura 3. Tercer Caso de Estudio, Viga Simplemente Apoyada

2 Estado del arte

2.1 Diseño paramétrico

En los proyectos de construcción gracias a los grandes avances tecnológicos se han desarrollado programas que permiten modelar estructuras en donde se facilita analizar y predecir el comportamiento de estas estructuras al estar expuestas a distintos factores, ya sea en cuanto a distintos diseños y distintas cargas actuando durante la vida útil de la estructura.

Al estar conceptualizando un proyecto en un modelo computacional es muy común que se soliciten cambios, ya sea en cuanto a dimensiones, ubicaciones de elementos, cargas actuantes y muchas veces modificar los modelos para adaptarlos a los cambios imprevistos toma mucho tiempo, además de que en ocasiones el flujo de trabajo se puede ver afectado tras realizar cambios sucesivos (Davis, 2013). Es por esto que con el tiempo y nueva tecnología se ha desarrollado una nueva forma para realizar los diseños, llamada diseño paramétrico. Desde una perspectiva histórica existe una controversia en la definición de diseño paramétrico (Davis, 2013). Por definición todos los diseños deberían considerarse paramétricos ya que tienen parámetros asociados que vienen de reglas que derivan sus resultados (Muttio, Botello, & Tapia, 2017).

El diseño paramétrico se basa en modelos que cuentan con parámetros variables en su configuración, esto permite al usuario manipular los parámetros de forma que la geometría conceptualizada cambie con los mismos. El objetivo del diseño paramétrico es obtener un modelo que sea lo suficientemente flexible para poder realizar cambios en él lo más rápido y con el menor coste (Davis, 2013).

Un modelo paramétrico requiere una mayor inversión de tiempo inicial comparado con un modelo no parametrizado (Muttio, Botello, & Tapia, 2017). De acuerdo con el autor Paulson, en la Figura 4 se muestra como una mayor inversión en las etapas tempranas de diseño de ingeniería conllevan a un menor costo en los cambios que se realicen posteriormente al diseño (Paulson, 1976).

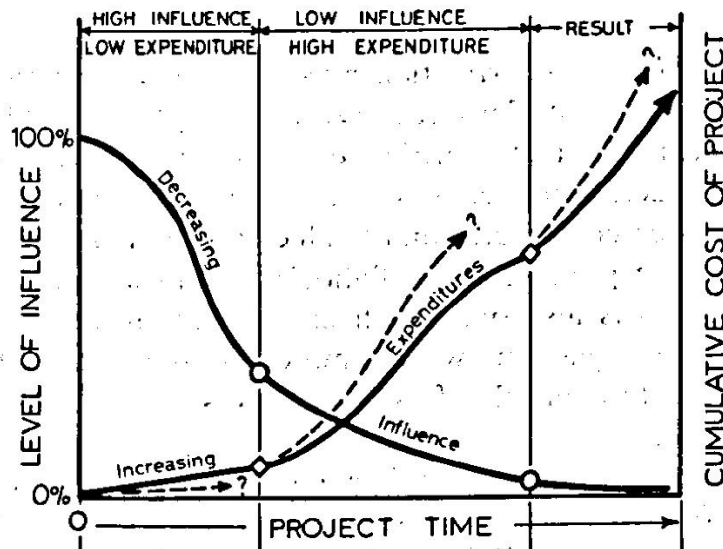


Figura 4. Curva de Paulson. Citado de (Paulson, 1976)

Los parámetros para modificar los modelos van desde el tamaño de la estructura, cantidad de elementos, longitud de los elementos, altura, áreas, funciones matemáticas para describir la forma del elemento y su ubicación, número de secciones en donde simplemente se puede modelar una sección de modelo y ajustar cuantas veces se requiere repetir la misma forma, hay una gran cantidad de parámetros que se pueden establecer y estos dependen tanto de las necesidades del proyecto como del proyectista. Una de las grandes ventajas del diseño paramétrico es que se pueden generar estructuras complejas utilizando funciones matemáticas como parámetros, la Figura 5 muestra una estructura modelada paramétricamente, esta figura muestra una estructura ya definida en su fase final, pero en su fase de diseño bien se podría cambiar los segmentos que tiene y su forma con tan solo modificar sus parámetros.

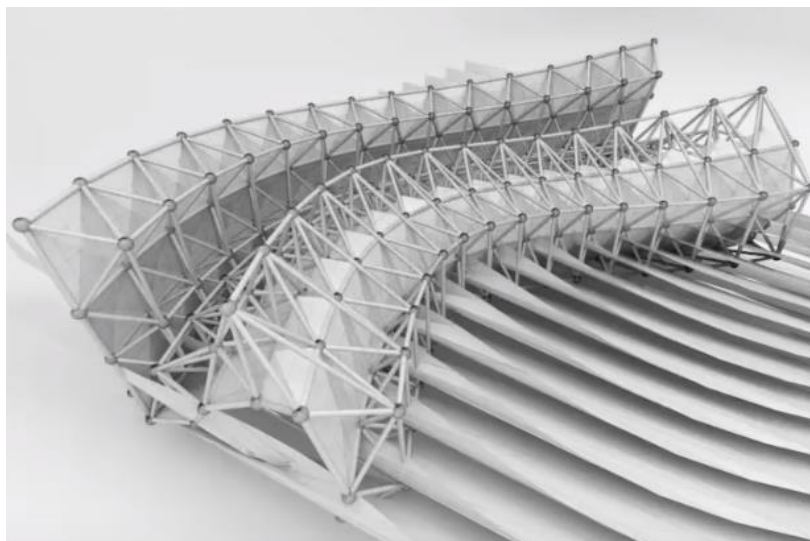


Figura 5. Pabellón Paramétrico. Citado de (Rodríguez, 2015)

La Figura 6 muestra una estructura modelada a través de diseño paramétrico utilizando programación visual en donde se permite ver con mayor claridad el proceso y la relación que existe entre los parámetros y el modelo obtenido. La programación visual permite crear programas de manera visual, utilizando nodos y cables (Gil & Parsons, 2014). En esta figura los parámetros que permiten modificar la estructura se encuentran en los rectángulos de color gris que en programación visual serían los nodos, se puede observar de manera clara las posibles soluciones estructurales existentes al modificar distintos grupos de parámetros mostrados en la imagen, por ejemplo, se puede obtener una estructura con orificios más abiertos, sin orificios, con mayor número de secciones y todo esto sin agregar nuevos elementos, simplemente modificando los parámetros existentes.

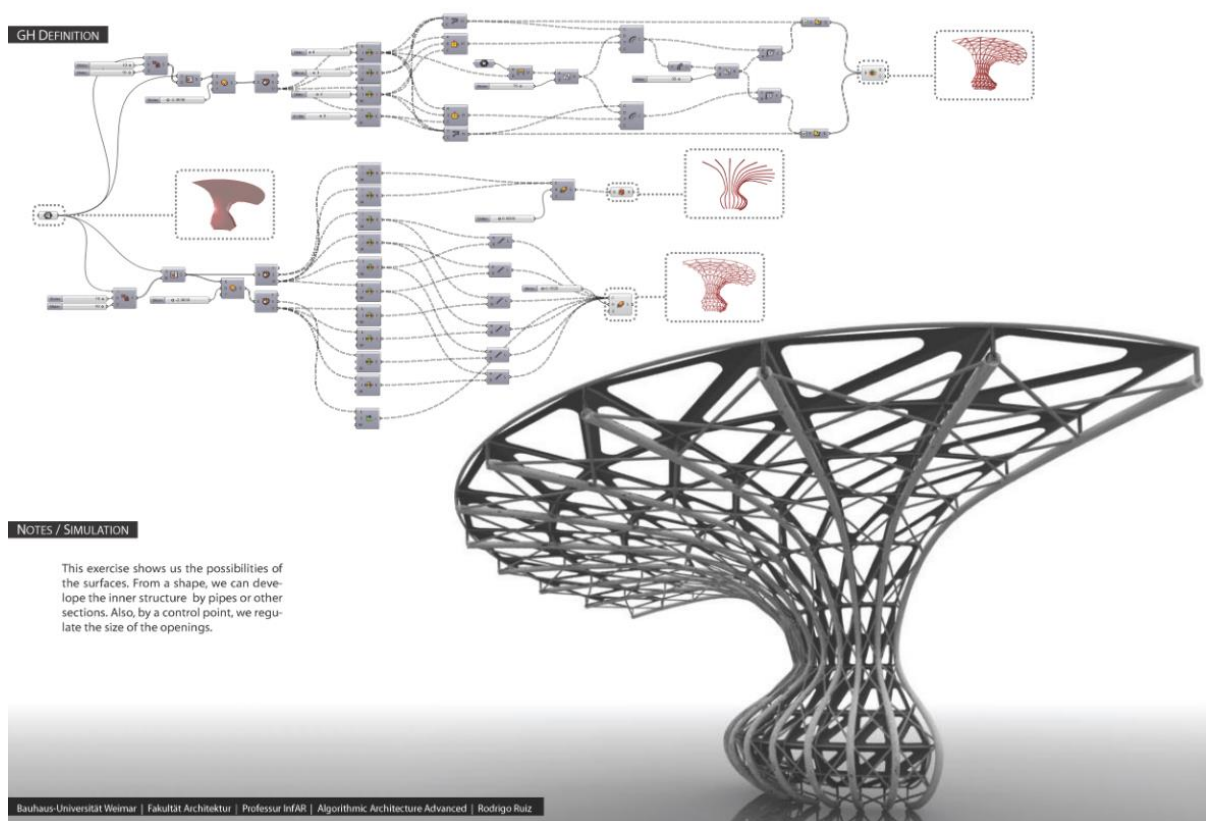


Figura 6. Distintas Soluciones Diseño Paramétrico. Citado de (Ruiz, 2015)

Se han hecho muchos trabajos e investigaciones respecto a diseño paramétrico de los cuales se consultaron algunos que se realizaron desde el año 2013 a la actualidad, lo que tienen en común estos trabajos es que en la mayoría se utilizan los mismo programas y que conforme avanza el tiempo se han realizado investigaciones más complejas utilizando diseño paramétrico, estos

hechos demuestran que los programas utilizados no han perdido vigencia con el tiempo y han mejorado con el tiempo permitiendo a los usuarios realizar modelos más complejos.

En 2013 Lee y Ha aprovecharon el diseño paramétrico para desarrollar la metodología llamada por sus singlas en ingles CIBIM (Customer Interactive Building Information Modeling). Su metodología consiste en que los clientes de unos departamentos puedan interactuar con el modelo realizado en Revit™ para distribuir las zonas interiores a su gusto y necesidad, ya sea agregar más habitaciones dependiendo el número de integrantes de la familia o colocar un área social más grande de acuerdo a su preferencia. La Figura 7 muestra los cinco pasos que se utilizan en la metodología. El primer paso consiste en generar muros fijos en donde el cliente no tiene control sobre su ubicación. El segundo paso consiste en generar muros móviles en donde el cliente ya tiene control y puede moverlos o quitarlos a su gusto. El tercer paso consiste en calcular el área de las habitaciones generadas, el área de calcula de manera automática ya que se encuentra parametrizado dependiendo de los muros existentes. El cuarto paso consiste en generar el techo y la ubicación de la iluminación, este paso también es automático ya que el modelo mantiene una conexión paramétrica entre la ubicación de los muros y la iluminación. El quinto paso coloca los muebles en sus posiciones paramétricamente siempre y cuando el cliente siga ciertas reglas al modificar el modelo (Lee & Ha, 2013).

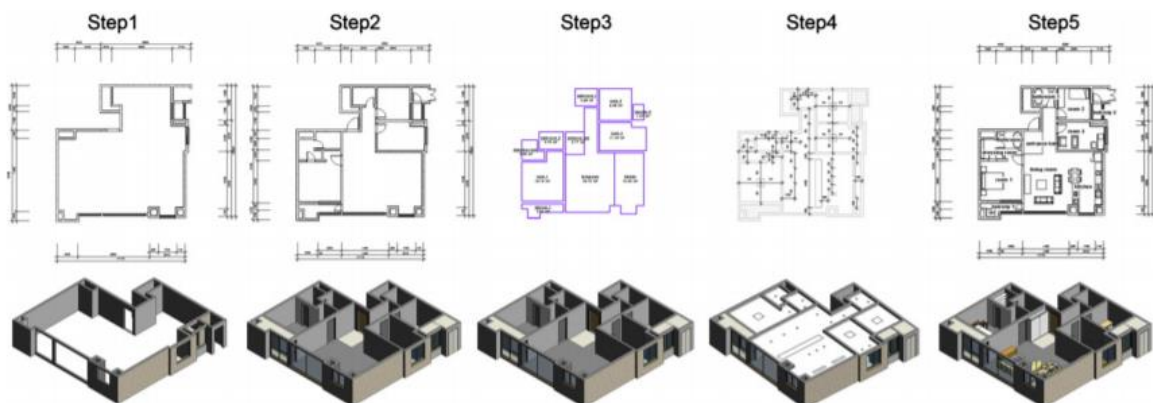


Figura 7. Pasos Método CIBIM. Citado de (Lee & Ha, 2013)

En 2014 Kensek utilizó el diseño paramétrico para generar ocho casos de estudio para experimentar con sensores que al recibir las condiciones ambientales como la luz solar se puedan adaptar y modificar la fachada para mejorar la experiencia del usuario o aprovechar al máximo la iluminación. De los ocho casos de estudio uno utilizó Arduino para percibir las condiciones ambientales, Rhino para realizar el modelo en tiempo real, Grasshopper para

realizar el modelo paramétrico a través de programación visual y por último se utilizó el componente Firefly para realizar la conexión entre los datos obtenidos del Arduino y los parámetros de Grasshopper. El resto de los casos de estudio utilizó de igual manera Arduino para recibir los datos externos, Revit como sustituto de Rhino, y Dynamo como sustituto de Grasshopper. Los resultados fueron satisfactorios, pero concluyeron que Grasshopper y Rhino tenían una respuesta más rápida al recibir los datos externos que Dynamo y Revit (Kensek, 2014). La Figura 8 muestra cómo se ajusta la fachada en Revit utilizando Dynamo al tener distinta intensidad de luz solar.



Figura 8. Resultado Facha Paramétrica Ante Distinta Luz Solar. Citado de (Kensek, 2014)

En 2016 Ibrahim demuestra con su investigación la gran versatilidad que hay en el diseño paramétrico, su investigación consistió en automatizar una persiana veneciana utilizando diseño paramétrico en Grasshopper en donde el parámetro a leer en tiempo real era la luz solar, con ayuda de los complementos Lady Bug y Honeybee para leer los ángulos e intensidad de la luz brindadas por un hardware permitió simular en que posición se debería ajustar la persiana para que la luz en el interior del edificio fuera la mejor y que no molestara a los usuarios. Concluyeron que una persiana automatizada de manera paramétrica brinda mejores resultados que una persiana común para el usuario (Ibrahim, Wagdy, Beccarelli, Carpenter , & Chilton, 2016).

En 2017 Muttio realizó un trabajo que tuvo como objetivo mostrar el desarrollo de una aplicación computacional con la capacidad de definir parámetros para estructurar una edificación, así obteniendo un campo de distintas soluciones en un lapso corto de tiempo, obtener una propuesta optima en cuanto a su resistencia mecánica, disposición topológica y que satisfaga las necesidades arquitectónicas solicitadas (Muttio, Botello, & Tapia, 2017).

En 2019 Pan realizó un trabajo para generar espacios multi funcionales y de grandes claros en arenas deportivas utilizando diseño paramétrico y optimización con multi objetivos. Para el recubrimiento en el techo consideraron tres tipos de estructuras, parrilla, marcos y celosías. El espacio interior y exterior de la arena cambia dependiendo la distancia promedio del espectador al centro de la arena, la distancia máxima de visión de diferentes actividades en la arena, ratios de puntos ciego y obstrucción, ratio de asientos utilizados durante actuación en escenario, ratio de asientos VIP con ángulo de visión horizontal al escenario y la acústica de la arena, al optimizar estos parámetros y teniendo la flexibilidad de ir actualizado automáticamente el modelo gracias a que fue modelado de forma paramétrica se genera el modelo más adecuado para cada caso. Concluyeron que los resultados obtenidos para un diseño arquitectónico son buenos, pero se requiere realizar más experimentación para generar modelos con cargas (Pan, Turrin, Louter, Sariyildiz, & Sun, 2019). La Figura 9 muestra algunos de los modelos que se generaron al modificar los parámetros de visión, acústica y peso estructural.



Figura 9. Modelos Generados con Distintos Parámetros. Citado de (Pan, Turrin, Louter, Sariyildiz, & Sun, 2019)

La Figura 10 muestra un mapa conceptual en donde se presentan los programas actuales que se utilizan en diseño paramétrica en el área de construcción, se divide en programas que generan diseño paramétrico a partir de programación visual y programas que no requieren de programación visual para realizar diseños paramétricos.

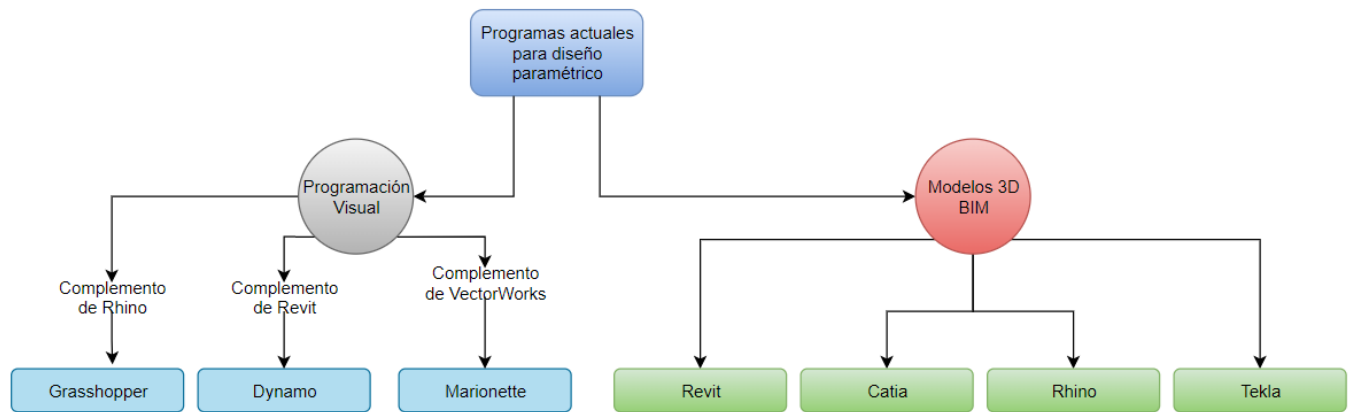


Figura 10. Programas Actuales para Diseño Paramétrico

2.2 Optimización estructural

La optimización estructural es un concepto fundamenta en ingeniería. Es imprescindible estudiar este concepto de manera profunda especialmente en épocas de incertidumbre y crisis económica. El diseño optimo busca adecuar a un problema de ingeniería la solución que satisface las restricciones de acuerdo con un criterio escogido entre menor costo, menor peso, facilidad de construcción, etc. (Cerrola & Annicchiarico, 1996).

Hablando de optimización estructural, existen diferentes métodos que se han desarrollado a lo largo de la historia de acuerdo con la tecnología y necesidades que se cuenta. Adicional a los métodos de optimización de acuerdo con el autor Bendsøe la optimización estructural se puede clasificar en tres categorías: tamaño (perfiles), geométrica y topológica (Bendsøe, 1989).

La optimización de tamaño como en la Figura 11 trata de determinar las medidas óptimas de los elementos que constituyen una estructura de forma predefinida, la optimización geométrica como en la Figura 12 trata de buscar la forma óptima de una estructura de topología fija y la optimización topológica como en la Figura 13 busca la distribución óptima del material en una estructura mediante la supresión de elementos y/o modificación de la conectividad entre distintos nodos (Sánchez, 2012).

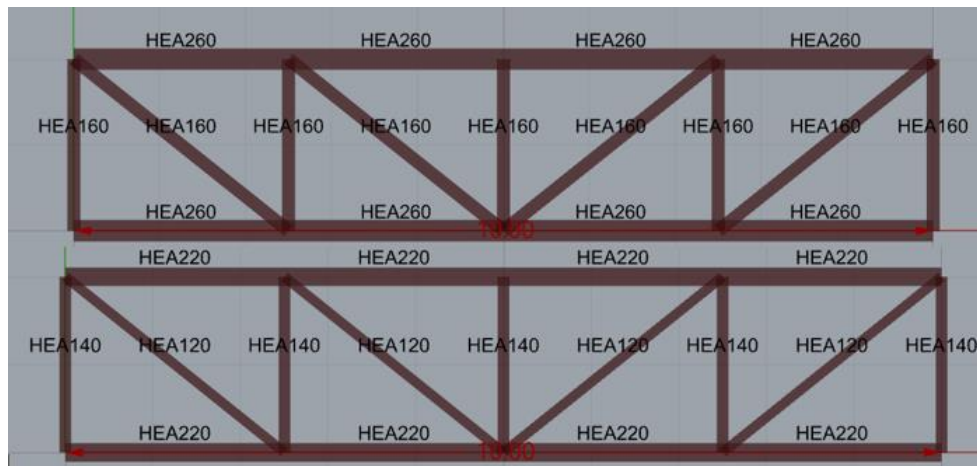


Figura 11. Ejemplo de Optimización de Tamaño

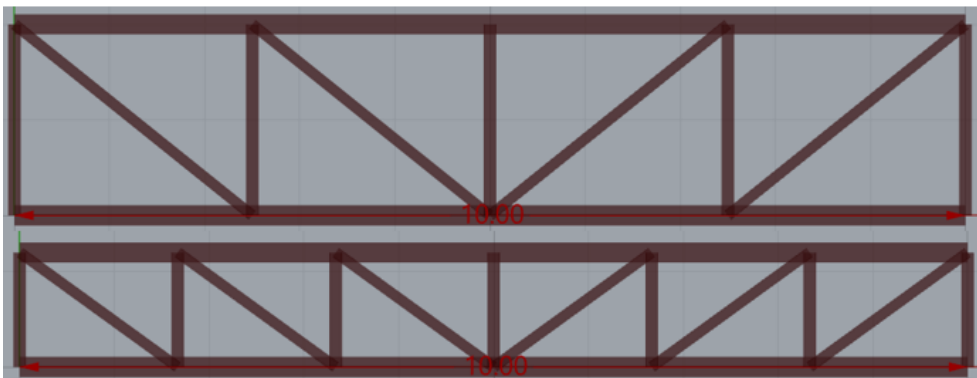


Figura 12. Ejemplo de Optimización de la Geometría

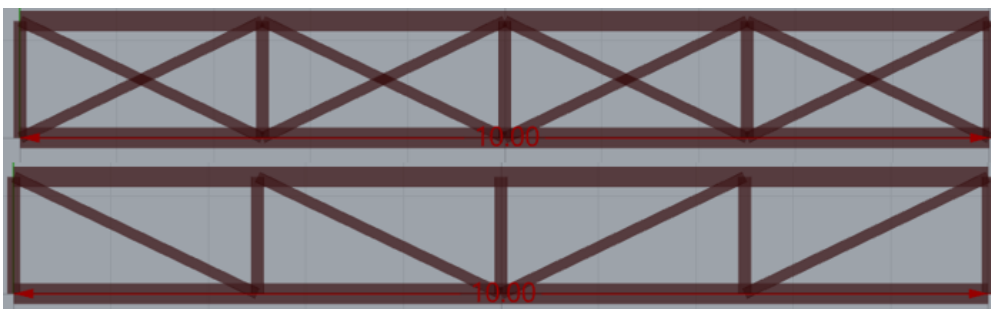


Figura 13. Ejemplo de Optimización Topológica

2.2.1 Optimización metaheurística

El termino metaheurística se mencionó por primera vez en 1986 por Fred Glover en su artículo sobre búsqueda del tabú, en un principio se rechazó que cambiara el termino heurístico por metaheurístico, pero esto dio pie a nuevas investigaciones sobre el tema. Para definir la optimización metaheurística primero se debe definir que es la optimización heurística el cual es un procedimiento que permite resolver problemas de gran complejidad de manera intuitiva, sabiendo esto la optimización metaheurística ejecuta procesos iterativos que permiten guiar y modificar las operaciones heurísticas para dar una mejor calidad a las soluciones, de igual manera toma como base procedimientos heurísticos para aplicarlo a distintos problemas (Sörensen, Sevaux, & Glover, 2018).

Samuel Sánchez indica en su tesis que las técnicas metaheurísticas comparado con las técnicas tradicionales de optimización (programación matemática, malla de puntos, técnica geométrica, técnicas híbridas), no funcionan a través de métodos o reglas preexistentes de búsqueda, aun así, las soluciones que proporcionan se consideran buenas y se obtienen en un periodo corto de tiempo. Tiene el inconveniente de no garantizar la obtención del optimo absoluto en los primeros resultados, esto hace necesario que el algoritmo tenga que llevar un proceso iterativo para llegar a la solución óptima. Estos métodos son utilizando cuando no existe algún algoritmo preestablecido que pueda optimizar el problema en cuestión. (Sánchez, 2012)

2.2.2 Algoritmos evolutivos genéticos

Los algoritmos evolutivos genéticos trabajan a partir de soluciones y se basan en los conceptos Darwinianos de supervivencia de dichas soluciones más aptas. Los diseños o parámetros se representan en forma de cadenas numéricas llamadas cromosomas en donde sus elementos se denomina genes, vienen siendo los parámetros a manipular para buscar las soluciones. Los cromosomas son generados y modificados de acuerdo a mecanismos naturales de evolución como la reproducción, el cruce y la mutación. Tras generar diferentes generaciones los diseños obtenidos representan los óptimos. Comparado con la técnica de programación matemática los problemas de optimización de estructuras a partir de tamaño y topología no restringen al método de optimización con algoritmo evolutivos. (Sánchez, 2012)

2.2.3 Investigaciones recientes sobre optimización estructural

Al hablar de optimización los trabajos realizados durante los últimos años tienen algo en común, y es que la mayoría se enfoca en utilizar algoritmos que obtienen resultados a través de iteraciones para lograr su objetivo. En el año 2012 Samuel Sánchez tuvo como objetivo en su tesis doctoral generar un algoritmo de optimización evolutiva en Ansys (Ansys, 1970) que no requiriera de una estructura predefinida para determinar y optimizar una geometría. Normalmente al momento de realizar una optimización estructural se requiere de parámetros de entrada que indirectamente predefinen una morfología inicial, para de esta manera no estar limitado por las preconcepciones del proyectista. Con el algoritmo generado el siguiente objetivo fue optimizar una estructura de diez barras y seis nodos en voladizo sujeto a restricciones de tensión y desplazamiento, al no necesitar una estructura predefinida para optimizar el modelo, el algoritmo solo requiere de la carga, número de apoyos y tipo de apoyos (Sánchez, 2012). La Figura 14 muestra los resultados obtenidos tanto de Sánchez, como de Ebenau y Balling en trabajos anteriores para comparar los resultados obtenidos a partir de distintos algoritmos. El algoritmo de Ebenau solo permite variar las coordenadas de los nodos superiores en el eje “y” por lo que el algoritmo desarrollado por Sánchez muestra un mejor óptimo y se demuestra como la manera de plantear el problema restringe el proceso de búsqueda impidiendo obtener mejores resultados (Sánchez, 2012).

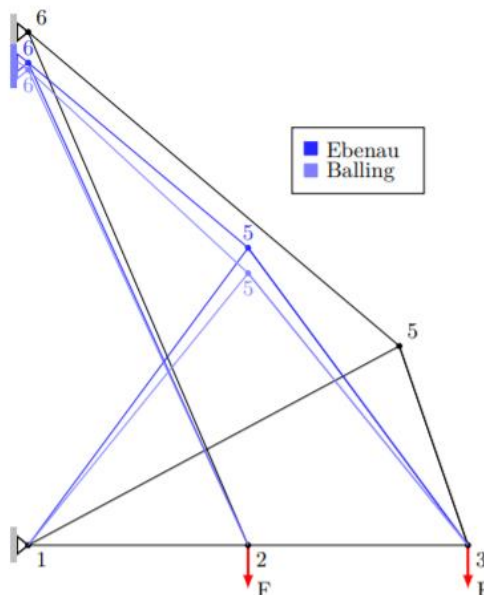


Figura 14. Resultados Obtenidos Por Diferentes Autores y Algoritmos. Citado de (Sánchez, 2012)

En 2014 Simonetti realizó una investigación para demostrar que los resultados obtenidos a partir de la optimización topológica utilizando el sistema de las siglas en inglés SESO (Smooth Evolutionary Structural Optimization) genera mejores resultados que el procedimiento denominado ESO (Evolutionary Structural Optimization) en estructuras de tensión plana. La teoría al pensar que el SESO genera mejores resultados que el sistema ESO viene de que si un elemento no es necesario realmente para la estructura su contribución en cuanto a rigidez estructural se va a ir reduciendo de manera gradual, por lo que remover este elemento se realiza de manera suave. Simonetti comprobó obtener mejores resultados con el método SESO para este tipo de estructuras y para estructuras de elasticidad bidimensional (Simonetti, Almeida, & Neto, A smooth evolutionary structural optimization procedure applied to plane stress problem, 2014). La Figura 15 muestra como el remover la barra gradualmente con un mayor número de interacciones en el método SESO tiene un menor efecto en las tensiones de la estructura, mientras que al remover algún elemento de manera radical con el método ESO genera mayores tensiones en la estructura.

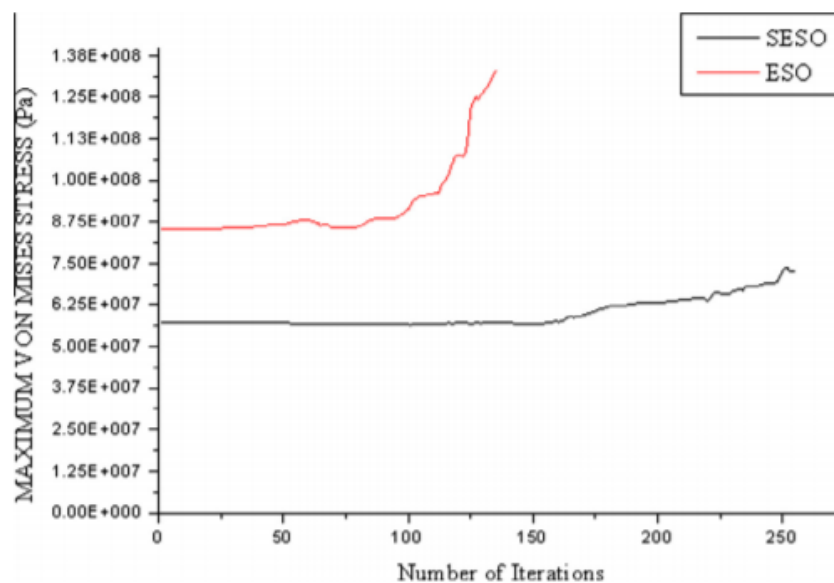


Figura 15. Comparación de Resultados Método SESO y ESO. Citado de (Simonetti, Almeida, & Neto, A smooth evolutionary structural optimization procedure applied to plane stress problem, 2014)

En 2018 Costanzi realizó una investigación sobre en el área de impresión de hormigón 3D. El tema de impresión 3D es de bastante actualidad y ha ido evolucionando de manera acelerada. En la actualidad existen moldes que se adaptan a la forma de la estructura que se desea imprimir, ya que si se desea imprimir una estructura con cierta curvatura esto sería imposible en un molde totalmente plano. Con lo anterior en mente el trabajo de Costanzi consistió en

optimizar la forma más adecuada del molde para la geometría a imprimir y de esta manera tener la menor deformación posible entre la forma de la geometría deseada y el molde. Para el proceso de optimización en el posicionamiento del molde se consideraron los siguientes pasos, primero se definió la geometría considerando las limitaciones físicas de la impresora que incluyen el mínimo radio de curvatura posible a imprimir y el ángulo máximo de inclinación para evitar deformaciones y deslizamientos en la estructura. El segundo paso consistió el colocar el modelo digital de la geometría a imprimir en un modelo digital del molde en Grasshopper y posteriormente utilizando el algoritmo evolutivo llamado Galápagos se optimizó la forma del molde considerando sus limitaciones reales para encontrar la forma más adecuada con la geometría (Costanzi, y otros, 2018). La Figura 16 muestra de forma esquemática el proceso realizado, en el paso uno en morado se encuentra la pieza seleccionada a imprimir, el paso dos muestra el molde adaptado con la forma adecuada para la estructura y el paso tres y cuatro muestra el proceso de impresión sobre el molde.

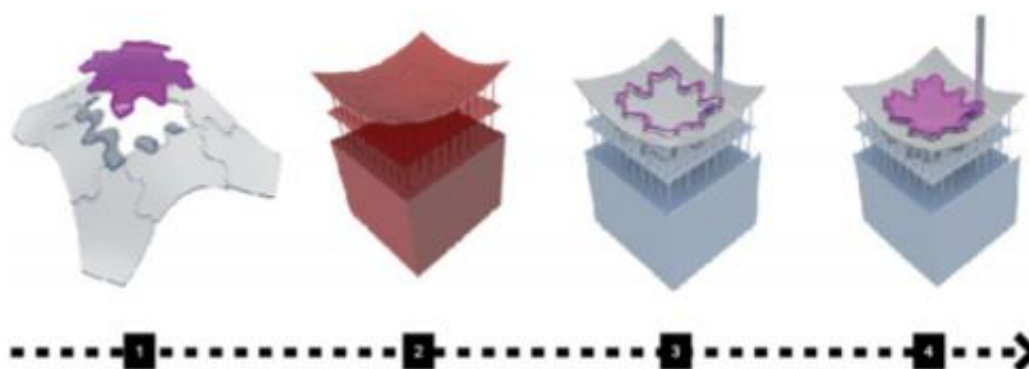


Figura 16. Proceso Propuesto de Impresión 3D de Concreto. Citado de (Costanzi, y otros, 2018)

En 2019 Simonetti escribió un artículo sobre optimización topológica combinando el método denominado por sus siglas en inglés BEM (Boundary Element Method) y el método ESO (Evolutionary Structural Optimization). La optimización topológica desarrollada funciona para problemas de elasticidad plana y se hace un acoplamiento entre el método BEM y ESO aplicando esfuerzos para representar las regiones que se deben remover durante la optimización topológica. En dicha investigación, se desarrollaron dos ejemplos, el primero fue una viga en voladizo en la Figura 17 y el segundo una viga simplemente apoyada en la Figura 18 (Simonetti H. L., Almeida, Leitao, & Neves, 2019).

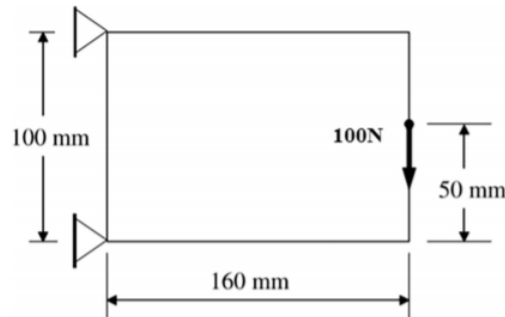


Figura 17. Viga en Voladizo Utilizada en Optimización Topológica. Citada de (Simonetti H. L., Almeida, Leitao, & Neves, 2019)

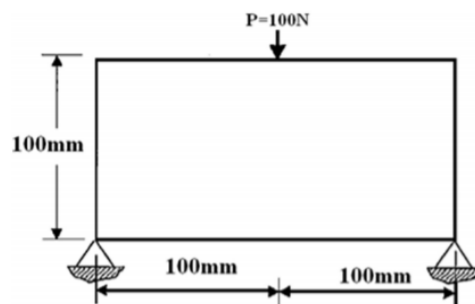


Figura 18. Viga Simplemente Apoyada Utilizada en Optimización Topológica. Citada de (Simonetti H. L., Almeida, Leitao, & Neves, 2019)

Las conclusiones que se presentaron fueron que los casos se analizaron para problemas elásticos lineales bidimensionales. Al combinar el método BEM-ESO se obtuvo una mejora considerable del costo computacional. Al calcular los esfuerzos con BEM es importante controlar las cavidades virtuales generadas para que no haya degeneración en el proceso de optimización topológica por lo que creen que en futuras investigaciones se pueden aplicar funciones de forma más sofisticadas o expresiones analíticas para calcular con los esfuerzos internos con una mayor calidad (Simonetti H. L., Almeida, Leitao, & Neves, 2019)

En 2019 Rombouts realizó un trabajo para proponer una manera de optimizar una estructura de parrilla tensionada y simplemente apoyada. Lo que se buscó fue optimizar la forma de la parrilla tomando y optimizando las fuerzas resultantes del proceso de instalación de la parrilla, de esta forma se puede disminuir los efectos de las cargas externas y desplazamientos sobre la parrilla. Para la optimización se utilizó el método llamado en inglés “moving asymptotes” que consiste en mover las asíntotas superiores e inferiores para ajustar la curvatura de las funciones aproximadas, es un método heurístico. El autor concluyó que el método de optimización

utilizado es efectivo especialmente para casos en donde las técnicas comunes para encontrar una geometría de la parrilla no funcionan como en casis de parrillas simplemente apoyadas o en voladizo. (Rombouts, Lombaert, De Laet, & Schevenels, 2019)

2.3 Investigaciones recientes sobre optimización estructuras y diseños paramétricos

Se habló sobre diseño paramétrico y también sobre optimización de las estructuras, en este punto se habla sobre optimización estructural realizada en software capaz de generar diseños paramétricos. En los trabajos consultados desde 2013 a la actualidad se observa una variable común y es que en la búsqueda de optimizar una estructura aplicaron diseño paramétrico, más que por facilidad en los cambios fue por necesidad ya que un modelo no parametrizado no permite evaluar las distintas opciones que se van generando durante un proceso de optimización. Como se mencionó anteriormente una de las ventajas que ofrece un modelo parametrizado es que permite la flexibilidad de optimizar la geometría de la estructura, de forma que durante cada proceso iterativo de la optimización se va generando un nuevo modelo de forma automática, optimizando también el tiempo del diseñador.

Em 2013 Kravanja utilizó el diseño paramétrico para optimizar una edificación industrial de una planta utilizando un modelo MINLP (Mixed-Integer Non-Linear Programming). MINLP permite generar un modelo y realizar una optimización a través de la modificación de distintos parámetros para lograr distintos objetivos como reducir el peso de la estructura cumpliendo con los requerimientos normativos (por ejemplo Eurocódigo) y optimizar la topología. Los parámetros para modificar en el modelo MINLP fueron la carga vertical de nieve, carga horizontal de viento, la luz de vano, la altura de los marcos principales de la edificación, tipo de sección transversal y la resistencia del acero estructural. Esta metodología empleada permite al diseñador decidir cuál es la estructura más conveniente al poder comparar gráficamente los parámetros obtenidos en cada posible solución (Kravanja, Turkalj, Silih, & Zula, 2013).

En 2015 Taleb utilizó ventaja de la optimización en el diseño paramétrico y generó un caso de estudio en la zona de Dubai denominada ‘Dubai Silicon Oasis’ su trabajo propone aprovechar el diseño paramétrico para tener mejores diseños y distribuciones en el diseño urbano para que las construcciones se perjudiquen lo menos posible por la radiación solar y así reducir el costo energético en la población. Los programas que utilizó fueron Grasshopper junto con Rhino

para modelar la zona paramétricamente, la Figura 19 muestra una sección del modelo donde se observan distintas edificaciones que se encuentran en la zona, estas edificaciones se representaron de manera sencilla con forma rectangulares. Para realizar las simulaciones ambientales se utilizó Ansys CFX. Grasshopper no tiene conexión directa con Ansys, pero ambos programas tienen la ventaja de poder aceptar modificaciones del usuario por lo que se utilizó un programa desarrollado en Python para realizar las operaciones automáticamente entre Grasshopper y Ansys. El ultimo programas utilizado fue Galapagos para optimizar las edificaciones, la Figura 20 muestra la posibilidad de modificar la altura, dirección y su morfología. Tras hacer distintas iteraciones se obtuvo una optimización en la radiación recibida en la zona desde un 1% hasta un 8%, no es un gran cambio y no se puede llegar y cambiar todos los edificios en una zona urbana para arreglar este tipo de problemas, pero para la actualidad en donde los problemas medio ambientales son muy grandes este trabajo demuestra que con las herramientas existentes generar este tipo de estudios en nuevas edificaciones y que es importante tener estos parámetros en cuenta durante un diseño (Taleb & Musleh, 2015).



Figura 19. Modelo Paramétrico. Citado de (Taleb & Musleh, 2015)

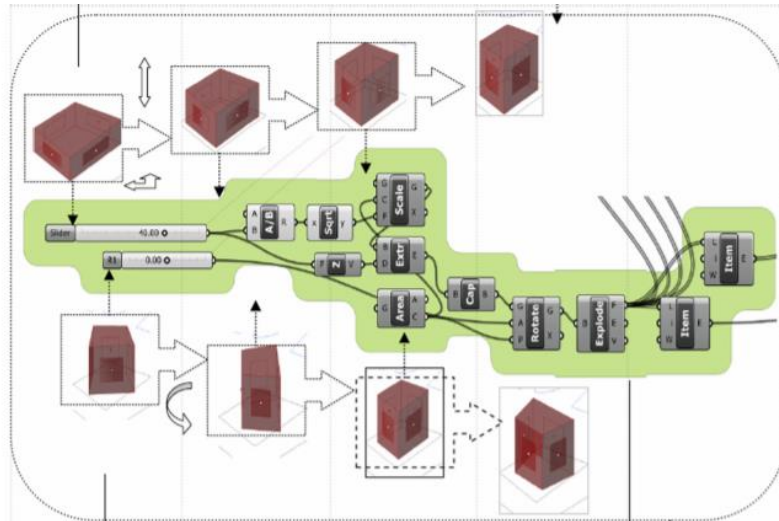


Figura 20. Parámetros de las Edificaciones. Citado de (Taleb & Musleh, 2015)

En 2018 Tomei se publicó un trabajo sobre la optimización estructural sobre los patrones de tipo diagrama en edificios altos (Tomei, Imbimbo, & Mele, 2018). El trabajo consistió en optimizar el peso del edificio intentando diferentes patrones en el diseño como un patrón de diagrama de 60°, 70° 80°, Angulo variable, doble densidad, densidad variable y líneas de esfuerzo ISO como se muestra en la Figura 21.

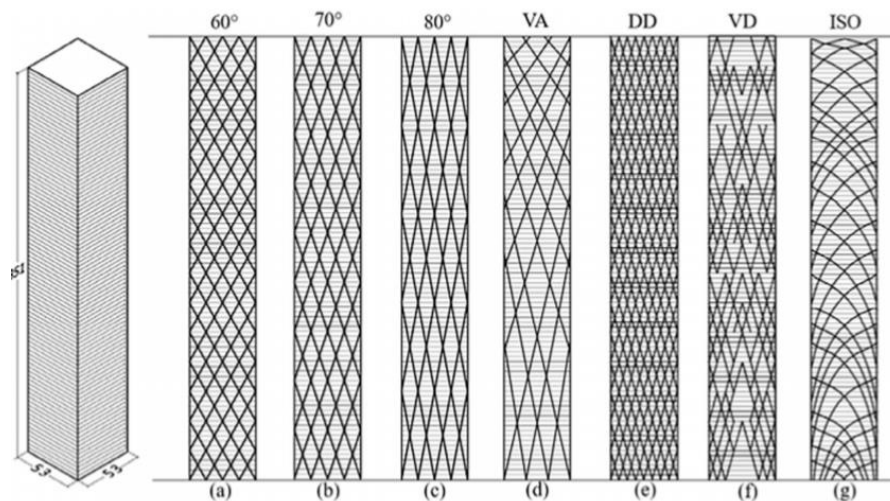


Figura 21. Posibles Patrones de Diagrama Durante Optimización. Citado de (Tomei, Imbimbo, & Mele, 2018)

El proceso se llevó de la siguiente manera se utilizó el complemento de Grasshopper para para realizar la geometría de forma paramétrica, se analizaron los resultados en Karamba 3D y se realizó la optimización con Galápagos probando los distintos patrones de la Figura 21 para

verificar los resultados (Tomei, Imbimbo, & Mele, 2018). La Figura 22 muestra la comparación de las derivas entre los diferentes patrones.

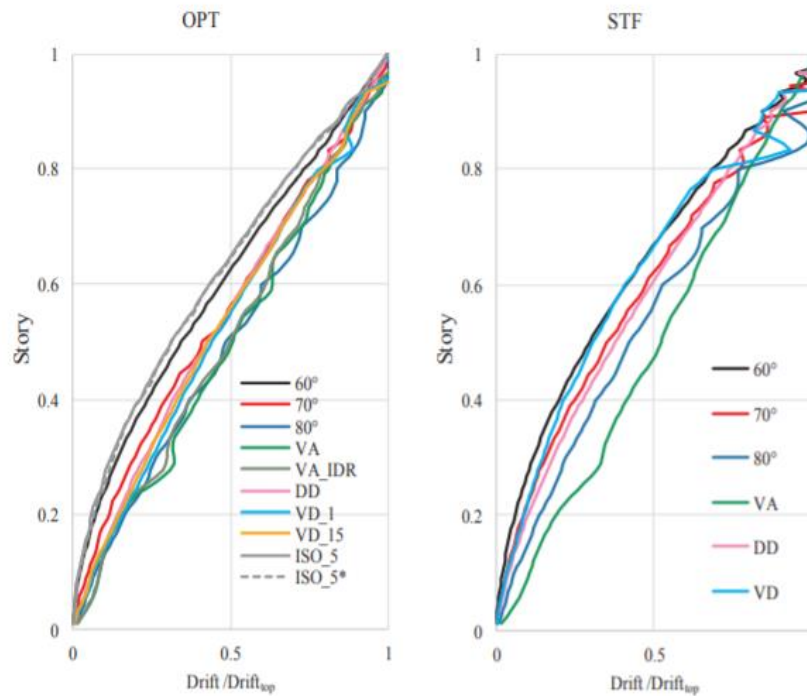


Figura 22. Comparación de Derivas entre los Patrones. Citado de (Tomei, Imbimbo, & Mele, 2018)

El objetivo de este trabajo fue obtener el patrón de diseño óptimo considerando el peso más bajo como lo óptimo, pero se dieron cuenta que no siempre el más pesado es el más barato y es por eso que se grafica el grado de complejidad en la construcción que es lo que podría elevar el costo del edificio además del peso. En este caso concuerda que los de menor peso son los de menor grado de complejidad. Comparando los resultados obtenidos por este método los resultados obtenidos fueron siempre de menos peso comparado con los resultados obtenidos a través de métodos tradicionales, por lo que se puede concluir que el método funciona de forma adecuada (Tomei, Imbimbo, & Mele, 2018)

En 2018 Yang propone un nuevo método de diseño exploratorio computacional. Se introdujo una formulación de optimización de problemas que permite cambiar la dimensionalidad del espacio objetivo y el diseño de exploración computacional puede reformular el problema de optimización usando análisis estadístico. Dentro de los criterios de optimización que se tomaron en se encuentran la geometría arquitectónica, clima y criterios relacionados a la estructura (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018).

Los programas que utilizaron fueron el modeFrontier (s.p.a, 2002) que tuvo dos funcionalidades, primero como base de datos sobre el modelo y los parámetros del modelo y segundo como algoritmo para realizar la optimización estructural. El segundo programa fue Rhino junto su complemento llamado Grasshopper, estos programas leyeron los datos de modeFrontier y generaron la geometría de la estructura. Aprovechando Grasshopper se utilizaron complementos con interoperabilidad llamados Daysim (Reinhart, 1998) que sirve para obtener simulaciones del comportamiento de la luz solar en el transcurso del día, utilizaron Energy Plus (EnergyPlus, s.f.) para hacer simulaciones de gastos de energía en la estructura por último se utilizó Karamba 3D para realizar la simulación estructural, todos estos parámetros se comparten entre estos complementos y modeFrontier para realizar el proceso de optimización (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018). La Figura 23 muestra un diagrama del proceso de comunicación entre el software que utilizaron. En la actualidad los programas que permiten una interoperabilidad con distintos complementos son de gran importancia ya que se puede trabajar en la misma interfaz de esta manera la comunicación entre los distintos complementos se encuentra garantizada. Grasshopper es uno de los grandes ejemplos en cuanto a interoperabilidad, McNeel desarrolló la página de internet llamada food4rhino en donde se puede obtener información sobre todos los programas y complementos relacionados con Rhino.

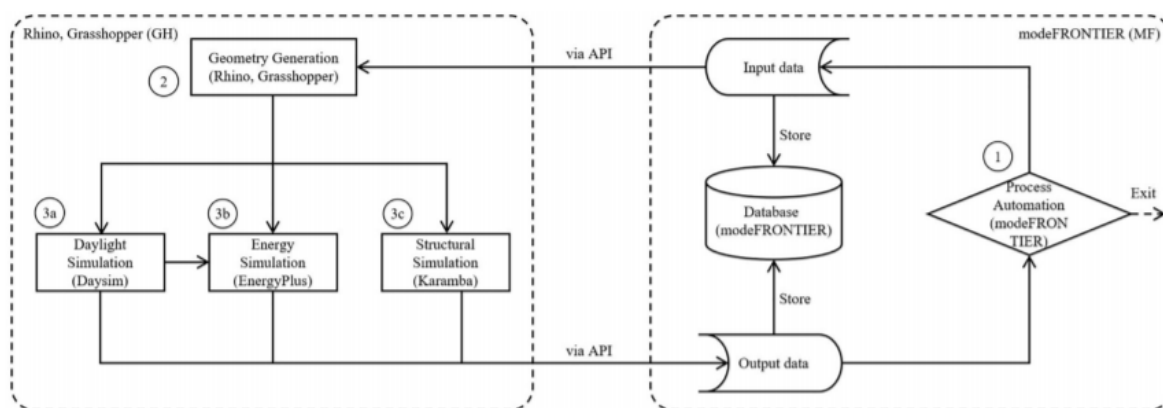


Figura 23. Software y su Proceso de Comunicación. Citado de (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018)

La optimización contiene dos fases de variables, la primera es un conjunto de diseños predefinidos del exterior del gimnasio observados en la Figura 24. El objetivo no es solo estructuralmente hablando sino obtener una estructura que sea capaz de trabajar junto con la luz ambiental para poder tener un gasto de energía durante la operabilidad más bajo que con otros diseños (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018)

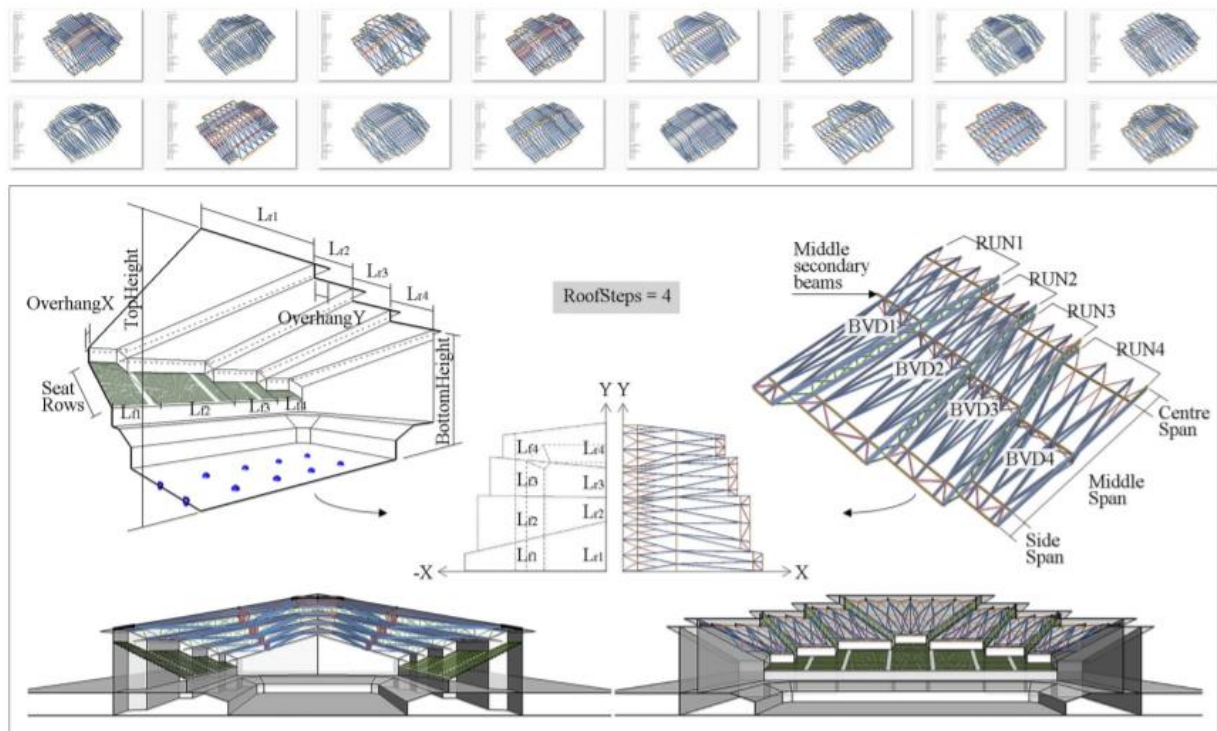


Figura 24. Parámetros Externos de Estructura Exterior Predefinidas. Citado de (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018)

Se concluyó que su diseño computacional exploratorio tubo buenos resultados para construcciones de gran escala como lo fue el gimnasio deportivo, la mejora que lograron frente a métodos de optimización tradiciones fue porque la formulación de del problema a optimizar puede variar gracias al diseño computacional exploratorio añadido. La formulación de optimización utilizado permitió la expansión en las dimensiones en el espacio objetivo, esto debe ya que el diseño computacional exploratorio borra o evita utilizar en el proceso de optimización y se enfoca en las variables que prometen mejores resultados. Comparado con un método tradicional se obtuvieron menos posibles opciones de diseño, pero todas con un igual o mejor resultado de optimización (Yang, Ren, Turrin, Sariyildiz, & Sun, 2018)

3 Herramientas

3.1 Rhinoceros 3D

Rhinceros 3D, también denominado como Rhino es un programa bajo licencia propietaria (GPL) que permite acceder a una versión de evaluación durante tres meses, esta herramienta permite crear, editar, analizar, documentar, renderizar, animar y traducir curvas NURBS, superficies, solidos, nubes de puntos y mallas poligonales (McNeel & Associates, 2019).



Figura 25. Logotipo de Rhinoceros 3D. Citado de (McNeel & Associates, 2019)

Las NURBS son representaciones matemáticas de geometría en 3D que son capaces de describir cualquier tipo de forma con precisión, abarca desde líneas, círculos, arcos o curvas 2D hasta los sólidos o superficies de forma libre en 3D, ejemplo Figura 26. La flexibilidad de los NURBS permite que se utilicen en cualquier proceso que va desde ilustrar y animar hasta el proceso de fabricar (McNeel & Associates, 2019).

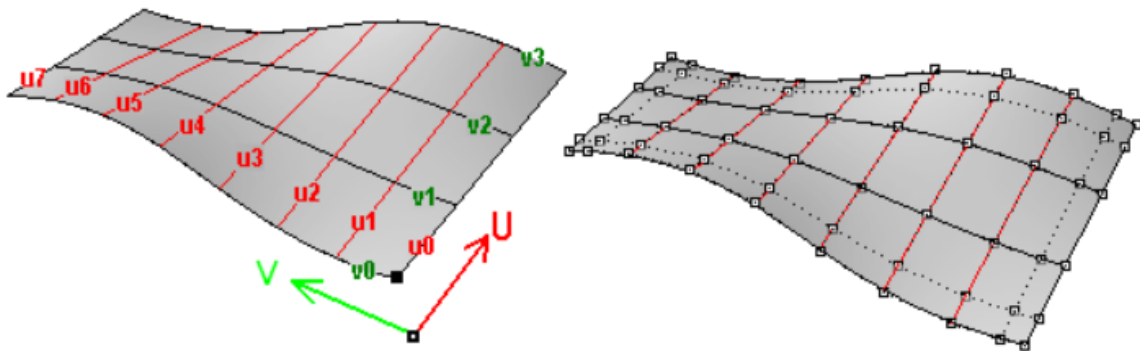


Figura 26. Ejemplo de uso de NURBS. Citado de (McNeel, 2019)

Además de poseer sus propias herramientas, Rhino contiene herramientas para desarrolladores que permiten una mayor flexibilidad con el usuario como lo son RhinoCommon (.NET), Grasshopper, rhino.Python, RhinoScript, administrador de licencias para plug-ins del Zoo y Rhino Installer Engine, Rhino posee código abierto lo que permite aún más a que el usuario pueda interactuar de manera personalizada con el programa (McNeel & Associates, 2019).

RhinoScript y Rhino.Python son potentes lenguajes de scripting de Rhino que gracias a su interoperabilidad con Rhino y sus bibliotecas, permite transmitir cualquier programa que se realice en estos entornos. La diferencia entre RhinoScript y Rhino.Python además del lenguaje de programación es que los programas que se desarrollan con RhinoScript solamente son funcionales con Windows, mientras que los programas desarrollados con Rhino.Python permiten utilizarse con Windows y Mac (McNeel & Associates, 2019).

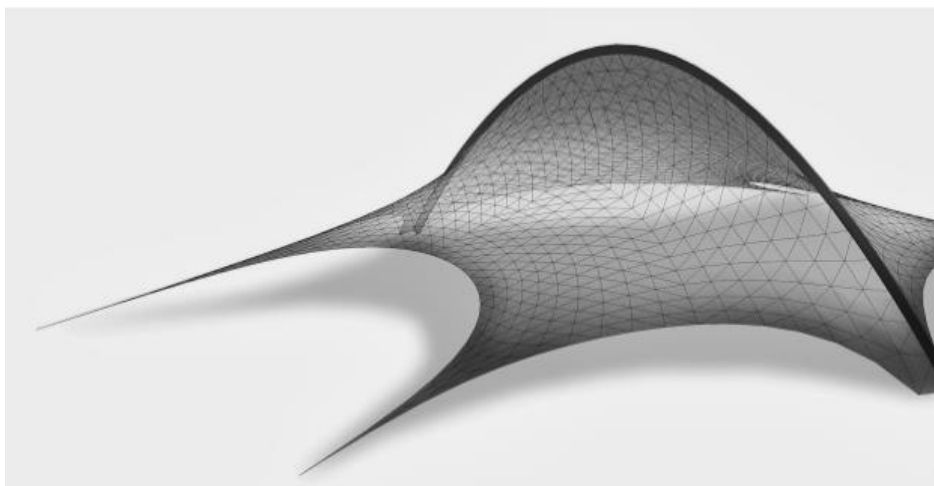


Figura 27. Estructura Generada en Rhinoceros 3D. Citado de (McNeel & Associates, 2019)

3.2 Grasshopper

Grasshopper es un editor de programación visual desarrollado por David Rutten en Robert McNeel & Associates. Grasshopper es un componente de Rhino3D que se puede utilizar por profesionales en diferentes campos de trabajo y estudio como lo es la arquitectura, ingeniería, diseño de productos y más. Grasshopper junto con Rhino ofrece la oportunidad al usuario de tener un control paramétrico sobre los modelos realizados, además de brindar una plataforma para desarrollar programas de alto nivel sobre una interfase grafica muy intuitiva (Gil & Parsons, 2014).



Figura 28. Logotico de Grasshopper. Citado (McNeel & Associates, 2019)

Al descargar la versión más actual de Rhino ya viene incluido el componente de Grasshopper pero también se puede obtener de Grsshopper3D.com para versiones más antiguas de Rhino. Grasshopper permite crear programas visuales denominados definiciones, estas definiciones están compuestas por nodos y cables (Gil & Parsons, 2014). La Figura 29 muestra la interfaz de Grasshopper.



Figura 29. Interfaz de Grasshopper. Citado de (Gil & Parsons, 2014)

Lo más común es desarrollar definiciones en Grasshopper con componentes denominados “slider” el cual es un componente que permite establecer valores numéricos de forma sencilla para el usuario y a partir de los valores generados y con las conexiones adecuadas dentro del programa se puede tener un control sobre distintas geometrías y de manera intuitiva. La conexión que tiene Grasshopper con Rhino es esencialmente en tiempo real, al ajustar los parámetros del “slider” se actualiza el modelo en Rhino como en la Figura 30 en donde variando el valor en el “slider” se incrementa o reduce el diámetro del círculo. Las definiciones pueden ser tan complejas como uno quiera y siempre hay distintas soluciones para llegar al mismo resultado como en la Figura 31 que muestra el mismo objetivo a través de código que se muestra en la ventana izquierda, este código se pueden desarrollar dentro de Grasshopper a través de componentes dedicados únicamente a desarrollar código en distintos lenguajes como Python, Visual Basic y C++ (Gil & Parsons, 2014). Grasshopper representa una puerta a muchos programas BIM como Tekla (Tekla, s.f.), permitiendo interoperabilidad entre ambos programas de forma sencilla. En el mercado existen entornos BIM como REVIT (Autodesk, s.f.) que cuenta con Dynamo (Autodesk, s.f.) que funciona de manera similar a Grasshopper.

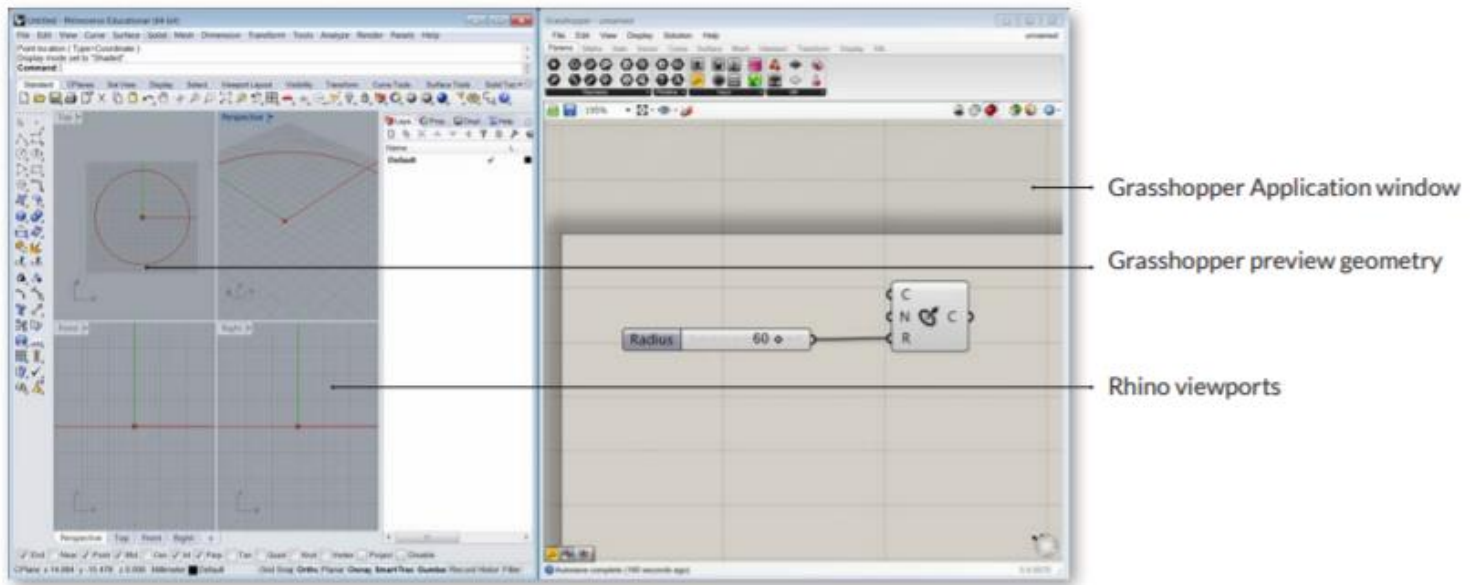


Figura 30. Programación Visual en Grasshopper y su Modelo Visual en Rhino, Citado de (Gil & Parsons, 2014)

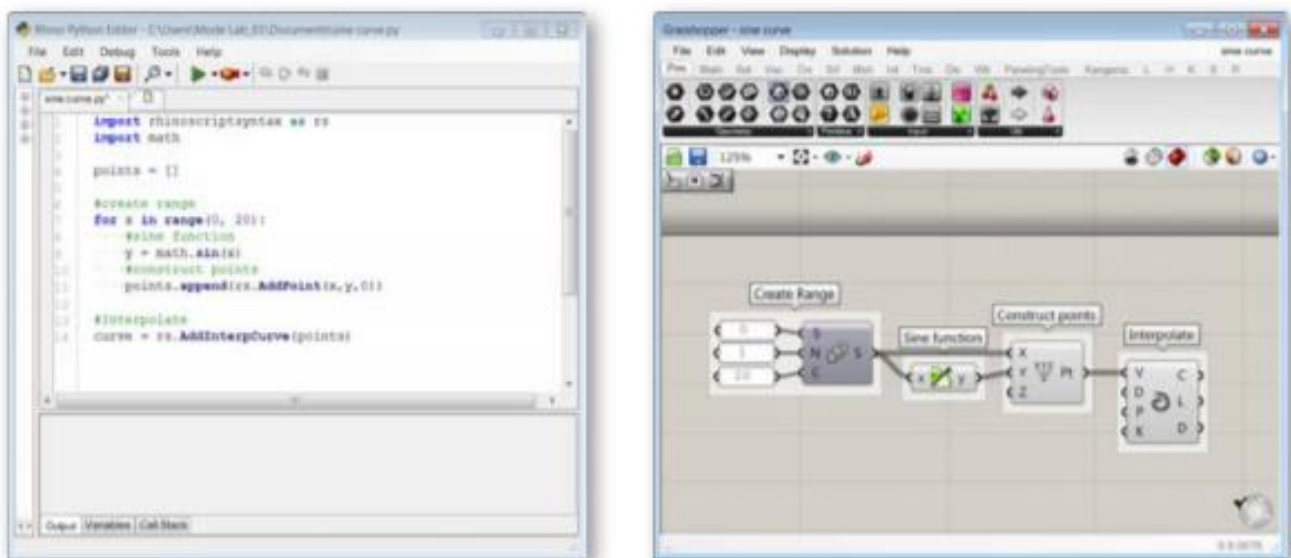


Figura 31. Proceso de Modelar una Curva a Través de Grasshopper en Distintos Métodos. Citado de (Gil & Parsons, 2014)

3.3 Karamba 3D

Karamba 3D es un programa de elementos finitos como muchos otros, pero tiene la particularidad de que se encuentra insertado en un entorno de programación visual como Grasshopper. Es sencillo de utilizar, se ha ajustado a las necesidades de ingenieros y arquitectos

en las fases temprana de diseño y trabaja de forma interactiva a un costo menor al resto de los programas (Preisinger, Linking Structure and Parametric Geometry, 2013).



Figura 32. Logotipo de Karamba 3D. Citado de (Preisinger, Karamba 3D Parametric Engineering, 2019)

Karamba 3D se encuentra completamente embebido en el ambiente paramétrico de Grasshopper, el cual es un complemento de la herramienta para modelar llamada Rhinoceros 3D. Al formar parte de esta conexión entre Grasshopper y Rhinoceros 3D se vuelve más sencillo combinar modelos con geometrías parametrizada, cálculos con elementos finitos y optimización con algoritmos como Octopus y Galápagos (Preisinger, Karamba 3D Parametric Engineering, 2019).

La Figura 33 muestra como se muestra Karamba 3D dentro del componente Grasshopper.



Figura 33. Karamba 3D dentro de Grasshopper. Citado de (Preisinger, Karamba 3D Parametric Engineering, 2019)

Grasshopper se encuentra orientado a objetivos y un ambiente de programación visual. Grasshopper provee de objetos como puntos, curvas, superficies, entre otros para computar la geometría. A continuación, se muestran 7 entidades agregadas a Karamba3D para construir los modelos estructurales (Preisinger, Karamba 3D Parametric Engineering, 2019):

- Modelo: Contiene la información relacionada a la estructura.
- Elemento: Puede ser viga, armadura, tipo Shell o resortes.
- Grupo de Elementos: Agrupa los elementos en el orden dado, los vuelve accesibles bajo un mismo nombre.
- Articulación: Define la conectividad entre los elementos vecindarios.

- Carga: Acción externa que se impone a la estructura.
- Sección Transversal: Define la sección de la geometría de los elementos.
- Material: Provee la información sobre el comportamiento físico del material provisto a la sección del elemento.
- Soportes: Define como se conecta la estructura con el suelo u otros elementos.

Los objetos de Karamba 3D se comportan como las entidades de Grasshopper que se pueden guardar en contenedores y pueden convertir en texto la información conectando los componentes como los paneles, de esta forma se pueden observar las propiedades fundamentales obtenidas durante sus procesos. Karamba 3D no requiere proveerse de toda la información descrita anteriormente, en caso de no contar con material por default se considera acero S235 de acuerdo con el Eurocódigo 3 (EN 1993-1-1, 2005), también el tipo de sección por default se considera como una sección circular de 114.4 milímetros de diámetro y espesor de 4 milímetros (Preisinger, Karamba 3D Parametric Engineering, 2019). La Figura 34 muestra el procedimiento simple para una viga en cantiléver programado en Karamba 3D.

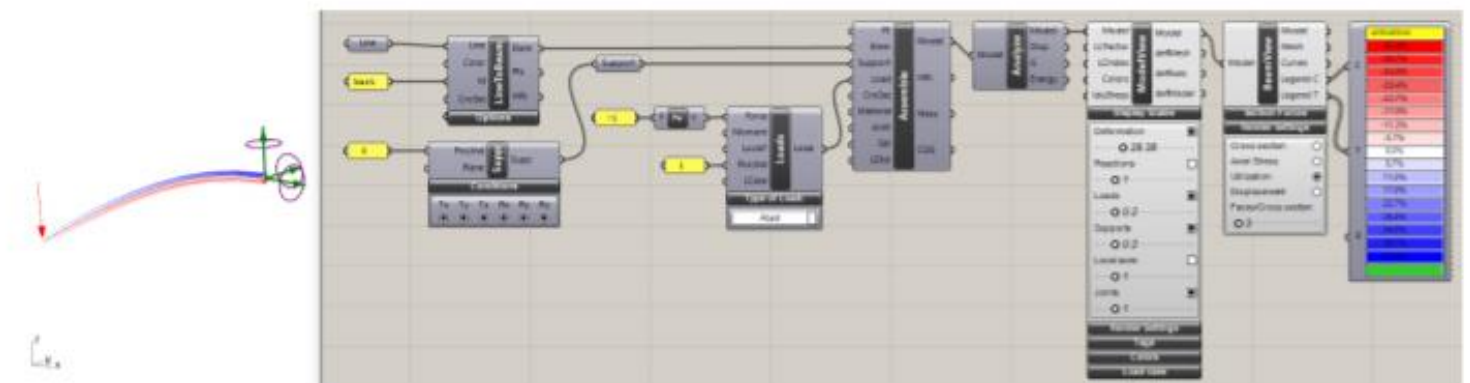


Figura 34. Viga en Cantilever Modelada Utilizando Grasshopper y Karamba 3D. Citado por (Preisinger, Karamba 3D Parametric Engineering, 2019)

3.4 Galápagos

Galápagos es un solucionador de algoritmos genéticos evolutivos de objetivo simple, este componente aplica lógica evolutiva para solucionar problemas específicos (Rutten, 2010). Galápagos es un componente que forma parte de Grasshopper, por lo que tiene una interacción completa con cualquier componente que se utilice dentro de Grasshopper.



Figura 35. Logotipo de Galápagos. Citado de (Rutten, 2010)

Como toda herramienta tiene sus pros y sus contras, alguna de sus desventajas es que como cualquier algoritmo evolutivo dependiente el proceso es lento, Si tiene procesos muy complejos puede llegar a tardar días o hasta semanas en correr el procedimiento. Otra desventaja es que no garantiza obtener una solución, por lo que se debe definir muy bien los parámetros a utilizar. Además de sus desventajas tiene grandes ventajas como que son muy flexibles ya que puede operar sobre diferente tipo de problemas, otra ventaja es que, aunque no se formulen muy bien los problemas a solucionar el algoritmo lo pueden procesar de manera adecuada ya que su proceso es progresivo. Una de las mejores ventajas es que tiene una gran interacción con el usuario, el proceso es muy transparente y se pueden encontrar subramas de buenos resultados óptimos (Rutten, 2010).

Para funcionar al algoritmo requiere de cinco partes (la mayoría ocurren de manera interna dentro de Galápagos) que se podría llamar como su anatomía (Rutten, 2010):

- Función “fitness” que brinda el objetivo al algoritmo.
- Selección del mecanismo.
- Algoritmo de acoplamiento.
- Algoritmo de coalescencia.
- Fábrica de mutación.

Un ejemplo corto de cómo funcionan los algoritmos evolutivos es el siguiente, la Figura 36 muestra el objetivo a optimizar con dos variables, que representan dos valores que pueden cambiar, en computación evolutiva las variables se denominan genes. Conforme se cambia el gene A el modelo mostrado puede mejorar o empeorar de acuerdo con su objetivo, pero para cada valor del gene A también se puede variar el gene B resultado en una mejor o peor combinación entre ambos parámetros. El objetivo del ejemplo es encontrar el pico más alto del modelo en la Figura 36. Cuando se coloca en un algoritmo evolutivo el algoritmo no sabe la forma por lo que su primer paso interno es poblar el modelo con una colección al azar de individuos o genomas, los genomas son valores que existen para cada gene, después de evaluar los resultados se obtiene el resultado en la Figura 37. Después de saber que tanto se acercan los resultados al objetivo se puede hacer una jerarquía entre los mejores y peores resultados, es entonces cuando el algoritmo “mata” a los genomas que no se adaptan haciendo referencia a términos evolutivos, por lo que los resultados restantes mostrados en la Figura 38 son los que mejor se adaptan al resultado. Lo que hace Galápagos es repetir el procedimiento hasta obtener un mejor resultado (Rutten, 2010).

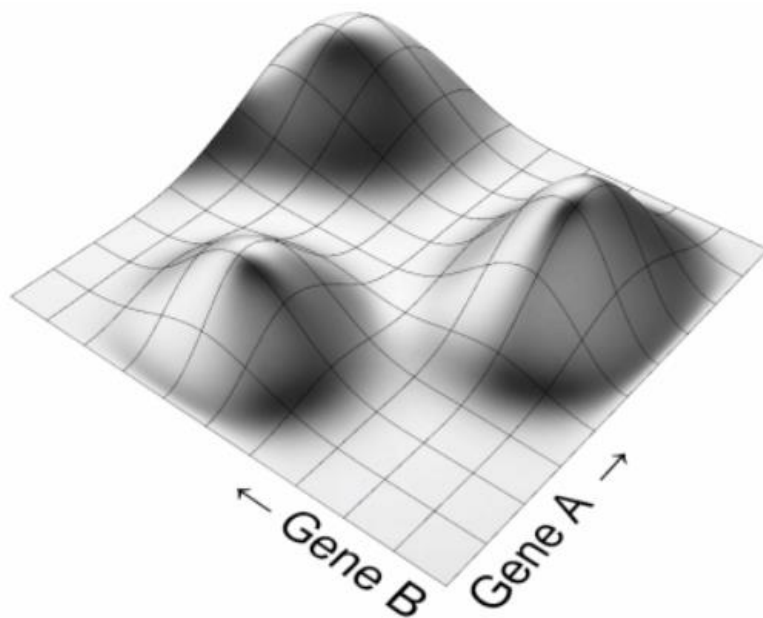


Figura 36. Modelo de Ejemplo Galápagos. Citado de (Rutten, 2010)

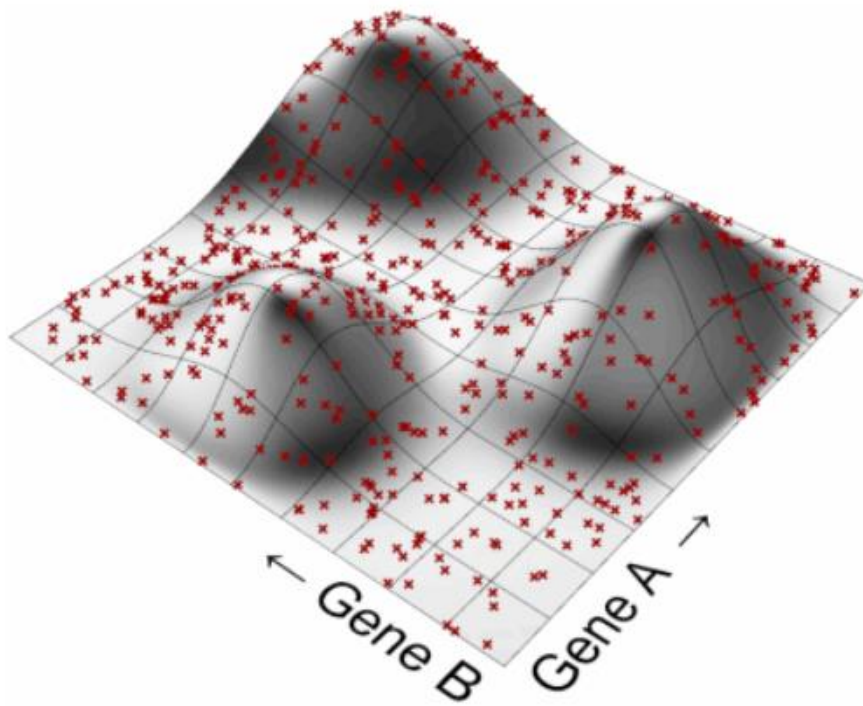


Figura 37. Población de Individuos del Ejemplo. Citado de (Rutten, 2010)

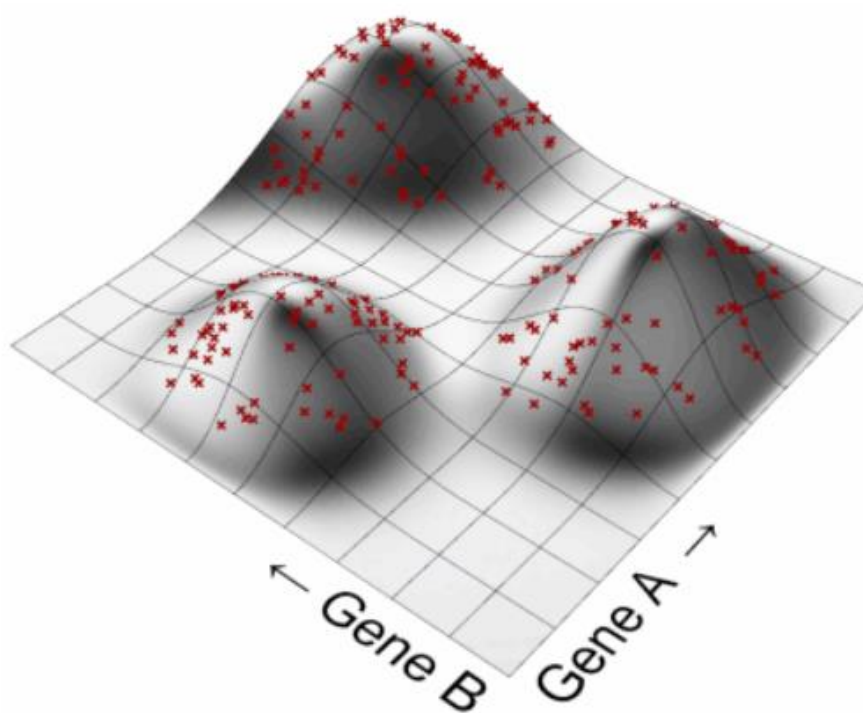


Figura 38. Limpieza de Genes Menos Aptos. Citado de (Rutten, 2010)

La Figura 39 muestra la interfaz de Galápagos en la pestaña opciones los parámetros en esta ventana más utilizados son el objetivo, también llamado fitness en donde se selecciona si el objetivo se busca minimizarlo o maximizarlo. Dentro de esta misma pestaña se pueden establecer límites de tiempo para las iteraciones.

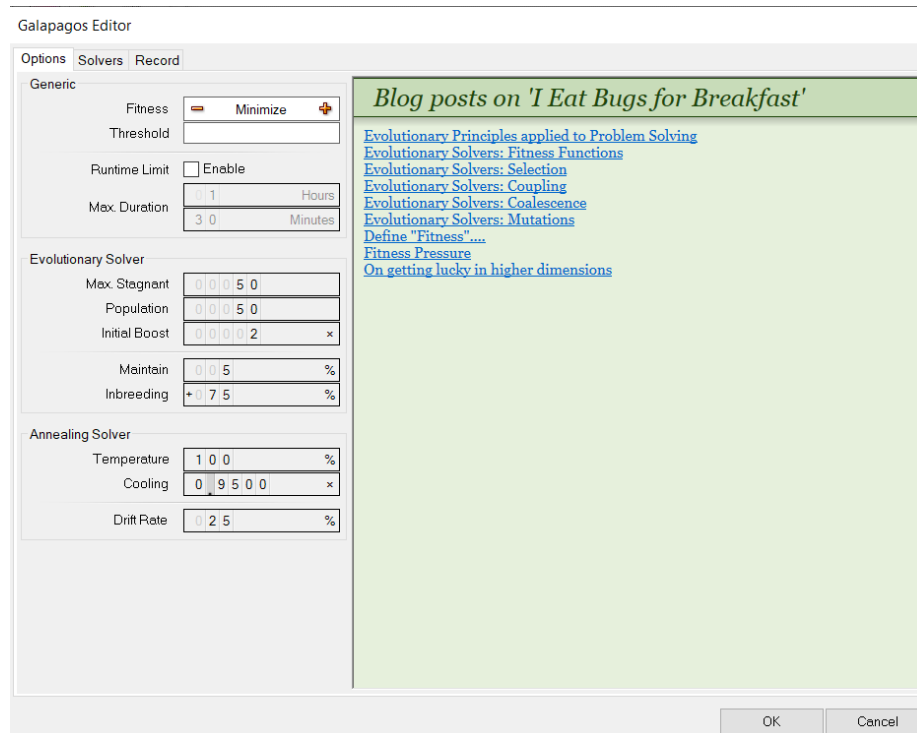


Figura 39. Interfaz de Galápagos Pestaña de Opciones

La Figura 40 muestra la pestaña llamada solvers dentro del interfaz de Galápagos, en esta sección de la interfaz es donde se corre el programa y se despliegan todos los resultados que se han obtenido durante el proceso iterativo en la ventana inferior derecha.

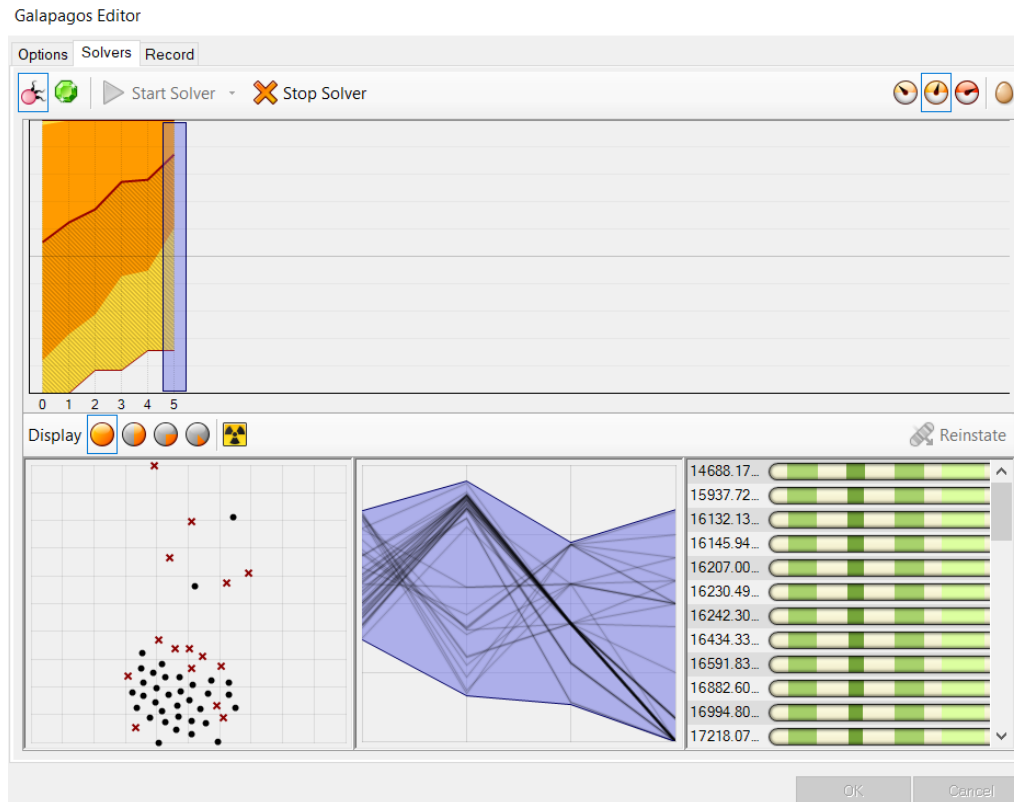


Figura 40. Interfaz de Galápagos Pestaña de Solvers

3.5 Octopus

Octopus es un componente para resolver problemas con algoritmos evolutivos orientado a diseño paramétrico (Vierlinger, 2012).



Figura 41. Logotipo de Octopus. Citado de (Vierlinger, 2012)

Al ser un algoritmo evolutivo, Octopus trabaja haciendo el mismo proceso lógico que Galápagos, la diferencia se encuentra en que Octopus introduce el principio de Pareto para poder trabajar a partir de múltiples objetivos (Vierlinger, 2012). Su interfaz se puede observar en la Figura 42, los cuadros que se grafican representan los resultados que se obtienen tras diferentes iteraciones y para el ejemplo de la imagen cada eje representa un parámetro que para este caso son distancia, flecha y número de columnas, a partir de estos parámetros se van situando los resultados de manera gráfica. Dentro de los ajustes que se ven del lado derecho se puede modificar la velocidad de las iteraciones, número máximo de iteraciones y el intervalo de tiempo.

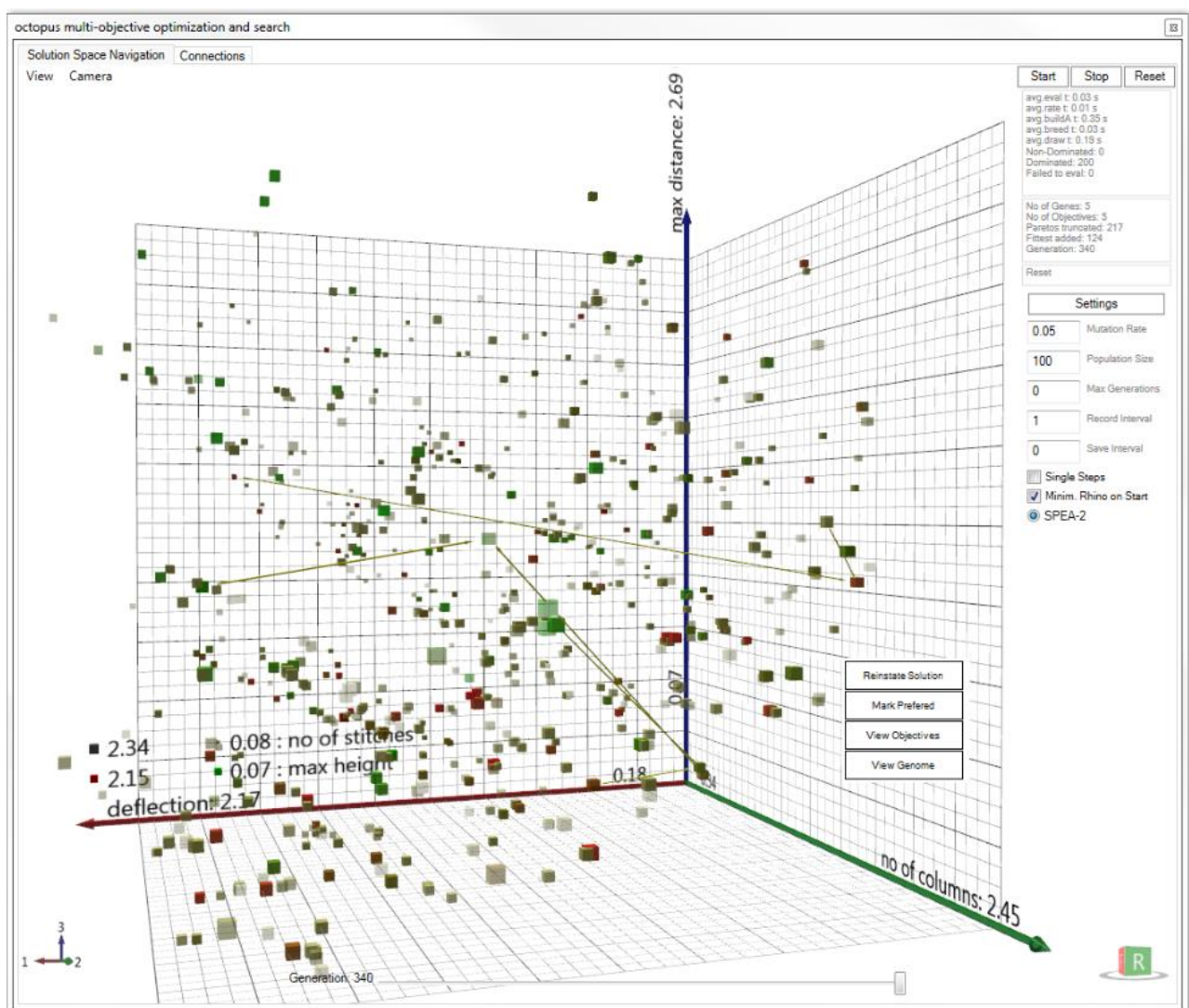


Figura 42. Interfaz de Octopus. Citado de (Vierlinger, 2012)

Parte de las tareas que puede realizar Octopus son las siguientes (Vierlinger, 2012):

- Obtener una diversidad de soluciones para un simple objetivo.
- Encontrar la mejor solución entre 2 a X objetivos.
- Mejorar soluciones por objetivos similares.
- Escoger las mejores soluciones durante la búsqueda.
- Cambiar objetivo durante la búsqueda.
- Soluciona modelos 3D para obtener retroalimentación visual.
- Guarda su historial de soluciones.
- Guarda todos los datos en el documento de Grasshopper.
- Exporta resultados a documentos de texto.

4 Casos de estudio

Ya se definió que es y como funciona el diseño paramétrico a través del programa Rhino y su complemento Grasshopper, se sabe la funcionalidad de los algoritmos genéticos de los complementos Galápagos y Octopus para optimizar y se conoce que el complemento Karamba 3D para análisis y verificación estructural, a continuación, se define la metodología aplicada y como se utilizaron los programas mencionados anteriormente para optimizar tres casos de estudio.

- El primer caso de estudio consta de una viga simplemente apoyada mostrada en la Figura 43 con una carga puntual a centro de la viga y una axial. Para este caso de estudio se generó un modelo paramétrico utilizando Rhino y su complemento Grasshopper. A partir de este modelo se utilizó el complemento de Grasshopper llamado Karamba 3D para realizar toda la simulación estructural en cuanto a cargas, material, perfiles y cumplimiento de verificaciones estructurales siguiendo el Eurocódigo. Para el proceso de optimización en este caso de estudio se utilizó un complemento de Karamba 3D que permite optimizar el perfil seleccionado a partir del Eurocódigo 3, obteniendo el perfil más ligero posible que cumpla con las cargas actuando en la estructura. Los perfiles utilizados durante esta optimización se obtienen de un catálogo de perfiles que se introducen de manera personalizada por el usuario.
- El segundo caso de estudio es una celosía empotrada mostrada en la Figura 66 con carga puntual al centro de la celosía. Para este caso de estudio se generó un modelo paramétrico utilizando Rhino y su complemento Grasshopper. Al igual que caso anterior se utilizó Karamba 3D para realizar la simulación estructural. En este caso el proceso de optimización fue diferente, primero se realizó una optimización topológica utilizando el componente de Karamba 3D llamado ‘eliminar componentes’ para eliminar elementos de la celosía que no se requieren y aligerar la estructura, posteriormente se utilizó el complemento de Grasshopper llamado Galápagos que a partir de algoritmos genéticos optimizó la estructura en dos fases, primero la geometría de la estructura modificando el peralte de la estructura y el número de secciones y posteriormente optimizando los perfiles de la estructura buscando obtener perfiles con altos ratios de utilización y menor peso de la estructura. En este mismo caso de estudio se realizó un proceso alterno en la optimización utilizando una combinación de los

complementos Octopus y Galápagos. En este proceso alterno el complemento Octopus se utilizó para optimizar el peralte y número de secciones de la estructura, mientras que Galápagos se utilizó para optimizar el tamaño de la estructura es decir el tipo de perfiles utilizados.

- El tercer caso de estudio es una viga armada simplemente apoyada con carga distribuida mostrada en la Figura 106. Para este caso de estudio se generó un modelo paramétrico utilizando Rhino y su complemento Grasshopper, pero a diferencia de los casos anteriores la geometría de la viga al tener parámetros variables por el usuario en cuanto a canto, longitud, espesores de patines y almas y ancho se utilizó un componente de Grasshopper de programación en código Python para modelar el perfil. Rhino cuenta con un entorno de clases, objetos y métodos llamado Rhino Geometry que a partir de comandos existentes en código facilita la conectividad entre el código en Grasshopper y el modelo visual de Rhino. Para este caso no se utilizó Karamba 3D para simulación estructural ya que al tener un perfil que no es estándar se dificulta la interacción con el modelo, por lo que se optó por utilizar otro componente Grasshopper de programación en código de Visual Basic, en este componente se generó un código para realizar las verificaciones estructurales a partir del Eurocódigo 3 y al mismo tiempo contiene las limitaciones necesarias para el proceso de optimización de la estructura. Para este caso de estudio se realizaron dos alternativas en el proceso de optimización, la primera constó en utilizar Galápagos para generar la viga armada modificando sus parámetros de canto, ancho de los patines y espesor de las placas en los patines y alma, todo esto mientras su objetivo consta de obtener el menor peso posible, cumplimiento de flecha y mayor ratio de utilización estructural de la viga. La segunda alternativa de optimización constó de utilizar el complemento Octopus en lugar de Galápagos modificando los mismos parámetros y buscando el mismo objetivo que la alternativa mencionada anteriormente.

4.1 Primer caso de estudio

En el primer caso de estudio se ha diseñado una viga simplemente apoyada mostrada en la Figura 43 considerando los siguientes parámetros:

- Luz por cubrir: 8 metros.
- Tipo de acero: S275.
- Carga puntual permanente al centro de la viga adicional al peso propio de 26 kN.
- Carga axial permanente de 75kN.
- Catálogo para optimizar el elemento limitado a perfiles HEB.
- Catálogo para optimizar el elemento con disponibilidad de perfiles HEB e IPE.

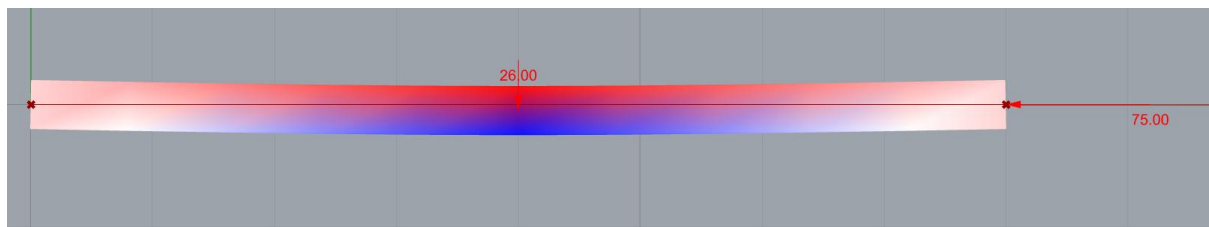


Figura 43. Viga Simplemente Apoyada (Primer Caso de Estudio)

El primer caso de estudio sirvió para verificar la funcionalidad de los complementos para optimizar perfiles de acero incluidos en Karamba 3D y al mismo tiempo analizar las limitaciones de este complemento.

4.1.1 Criterio para mayoración de cargas

Las cargas se mayoraron de acuerdo con la tabla A1.2 que se muestra en Eurocódigo y en la Figura 44 (EN 1990, 2002).

Persistent and transient design situations	Permanent actions		Leading variable action (*)	Accompanying variable actions	
	Unfavourable	Favourable		Main (if any)	Others
(Eq. 6.10)	$\gamma_{Gj,sup} G_{kj,sup}$	$\gamma_{Gj,inf} G_{kj,inf}$	$\gamma_{Q,1} Q_{k,1}$		$\gamma_{Q,i} \psi_{0,i} Q_{k,i}$
(*) Variable actions are those considered in Table A1.1 NOTE 1 The γ values may be set by the National annex. The recommended set of values for γ are : $\gamma_{Gj,sup} = 1,10$ $\gamma_{Gj,inf} = 0,90$ $\gamma_{Q,1} = 1,50$ where unfavourable (0 where favourable) $\gamma_{Q,i} = 1,50$ where unfavourable (0 where favourable) NOTE 2 In cases where the verification of static equilibrium also involves the resistance of structural members, as an alternative to two separate verifications based on Tables A1.2(A) and A1.2(B), a combined verification, based on Table A1.2(A), may be adopted, if allowed by the National annex, with the following set of recommended values. The recommended values may be altered by the National annex. $\gamma_{Gj,sup} = 1,35$ $\gamma_{Gj,inf} = 1,15$ $\gamma_{Q,1} = 1,50$ where unfavourable (0 where favourable) $\gamma_{Q,i} = 1,50$ where unfavourable (0 where favourable) provided that applying $\gamma_{Gj,inf} = 1,00$ both to the favourable part and to the unfavourable part of permanent actions does not give a more unfavourable effect.					

Figura 44. Coeficientes de mayoración. Citado de (EN 1990, 2002)

Para estado límite último las cargas se mayoraron por 1.35 mientras que para el estado límite de servicio las cargas se mayoraron por el coeficiente de 1.0.

4.1.2 Modelación paramétrica

En la Figura 45 se puede observar la programación gráfica que se definió en Grasshopper para obtener la geometría de la viga, así como el resultado obtenido al trasladar el código a un modelo en Rhinoceros 3D en la Figura 46. Al definir el primer punto en 0 metros en un eje X y el segundo punto en 8 metros en el eje X posteriormente se trazó una línea entre los dos puntos.

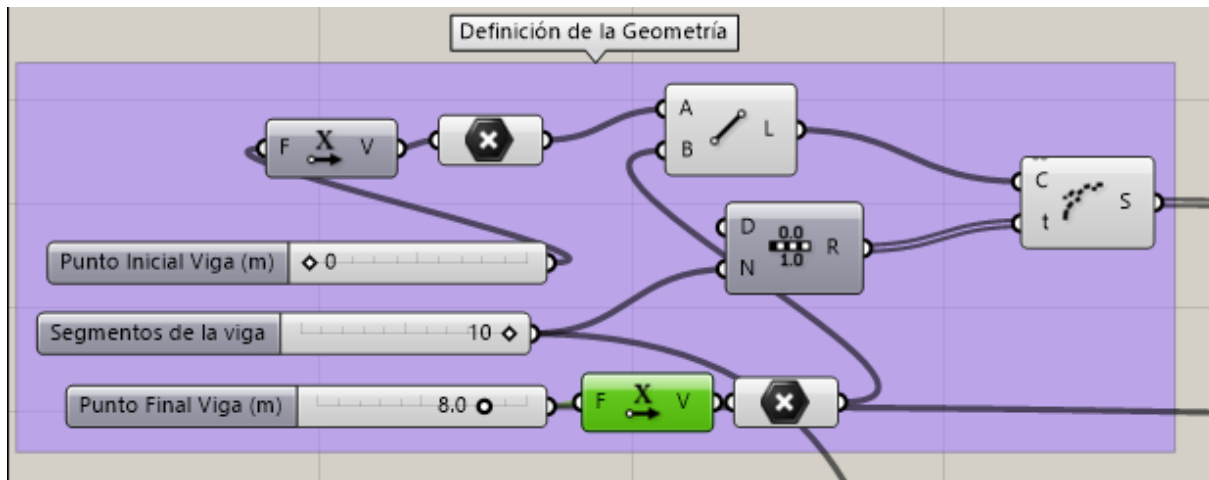


Figura 45. Definición Geométrica de la Viga Programada

Al deslizar los valores dentro de los parámetros de punto inicial y punto final mostrados en la Figura 45 es como se puede modificar el modelo de forma instantánea.

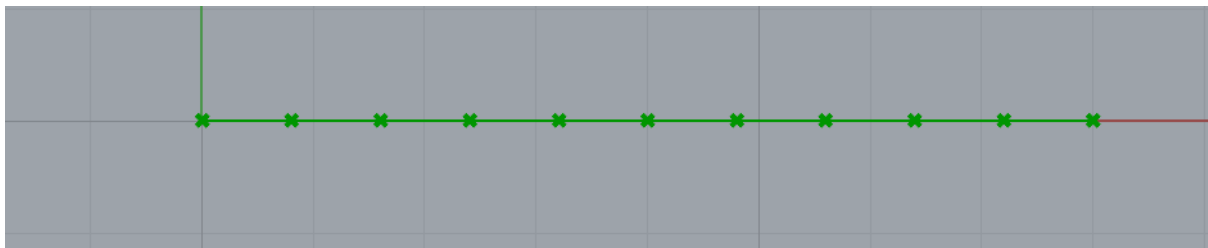


Figura 46. Modelo Visual de la Geometría de la Viga

4.1.3 Simulación estructural

Para realizar una comunicación entre la geometría programada en Grasshopper y Karamba 3D se utiliza el complemento de la Figura 47 el cual ayuda a definir el elemento como una geometría estructural, realizando una comunicación entre Grasshopper y Karamba 3D.

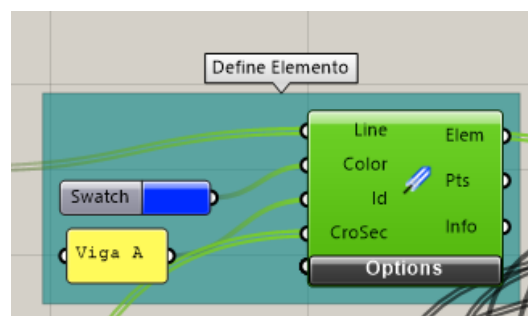


Figura 47. Definición del Elemento en Karamba 3D

Al definir los elementos los nodos se numeran de izquierda a derecha como en la Figura 48.



Figura 48. Definición de los Nodos en el Elemento

4.1.4 Definición de los apoyos

Con nodos ya definidos se procedió a definir los apoyos, colocando un apoyo simple en el nodo 0 y otro apoyo simple en el nodo 10 mostrado en la Figura 49. Los nodos se encuentran parametrizados de tal forma que si se cambia la longitud de la viga el nodo se actualizará de forma automática.

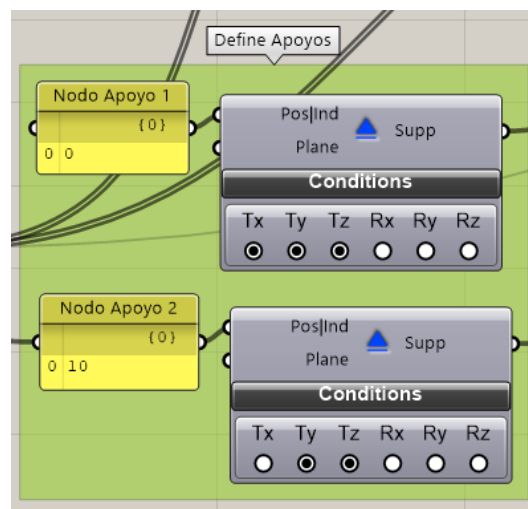


Figura 49. Definición de Apoyos

4.1.5 Definición de las cargas

Las cargas se definieron de acuerdo con la Figura 50. Estas cargas, su posición y dirección se encuentran descritas en la Tabla 1. Es importante aclarar que una de las limitantes que enfrenta Karamba 3D es que no se pueden realizar los casos de carga automáticamente por lo que se deben de introducir de forma manual, es decir que se debe describir el número de caso de carga e introducir el valor que tendrá la carga en ese caso. En la Tabla 2 se describió el caso de carga

utilizado para estado límite de servicio y estado límite último, así como sus factores de amplificación.

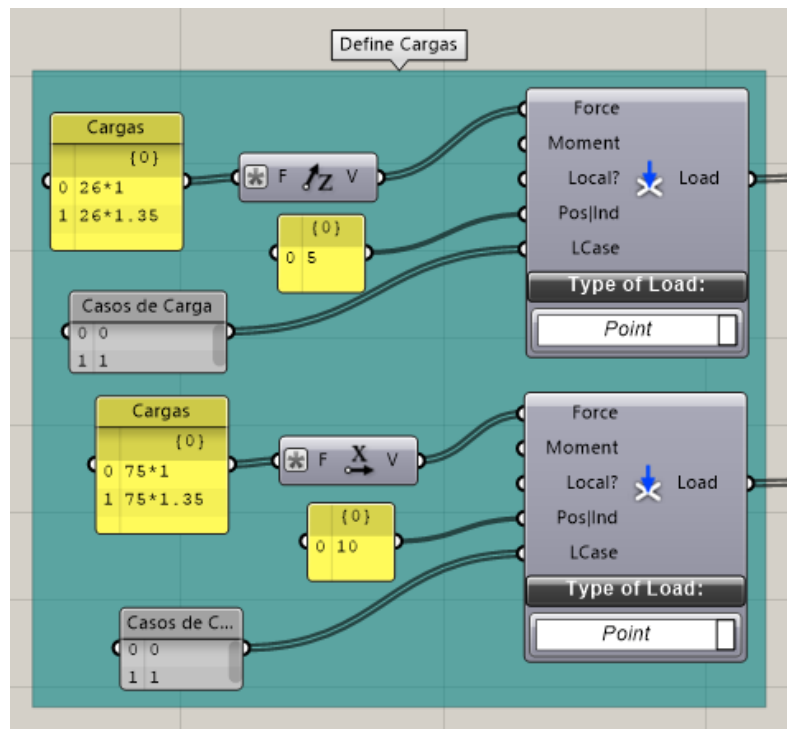


Figura 50. Deifnición de Cargas en Karamba 3D

Tabla 1. Magnitud y Posición de Cargas

	Nodo 5	Nodo 10
Carga (kN)	26	75
Eje y Dirección	-Z	-X

Tabla 2. Caso de Carga y Factores de Amplificación

	Estado Límite de Servició	Estado Límite Último
Caso de Carga	0	1
Factor de Amplificación	1	1.35

4.1.6 Definición de materiales

El tipo de acero, así como el tipo de perfiles y cantidad que se tomaron en cuenta en el modelo se programaron de acuerdo con la Figura 51. El acero utilizado es S275 y en este caso se limitó el catálogo de perfiles a secciones I de la familia HEB, con un total de 24 perfiles que abarcan desde el HEB 100 hasta el HEB 1000.

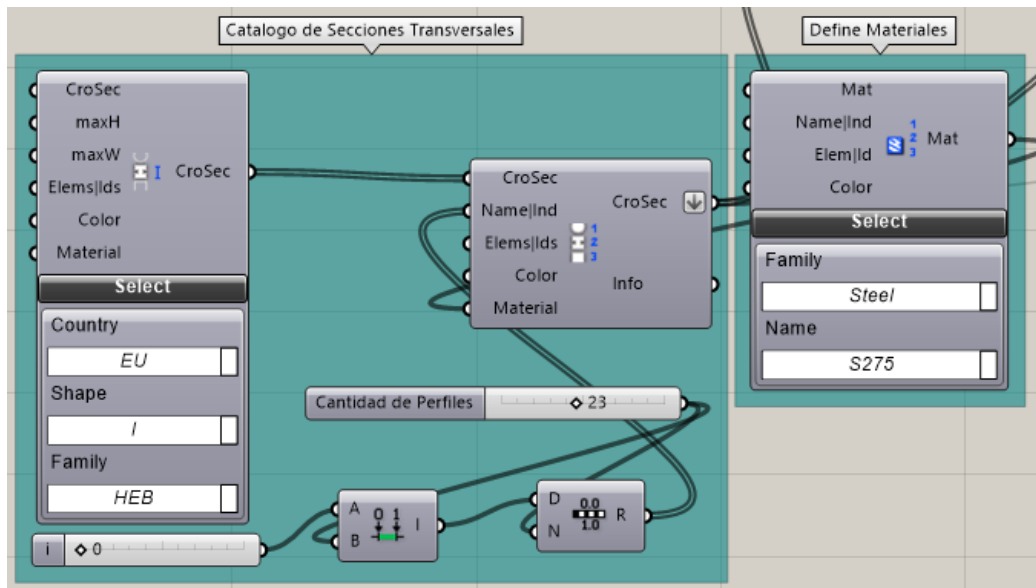


Figura 51. Catálogo de Perfiles de Acero

4.1.7 Ensamble del modelo

Con la geometría, los apoyos, la carga y los materiales definidos se ensamblaron en el complemento de Karamba 3D de la Figura 52. Este complemento traduce todo lo que se programó anteriormente para formar un modelo de elementos finitos que se puede manipular en Karamba 3D.

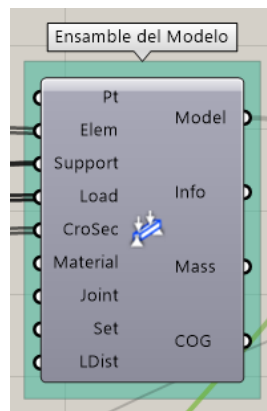


Figura 52. Ensamble del Modelo

4.1.8 Análisis estructural y su resultado

El análisis estructural se realizó con el complemento de análisis en la Figura 53. Este complemento obtiene los resultados utilizando teoría de primer orden, dependiendo del modelo se pueden utilizar complementos más potentes.

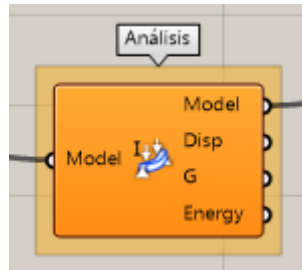


Figura 53. Análisis Estructural del Elemento

Para observar el modelo y sus resultados se utilizaron los complementos desplegados en la Figura 54. El primero es el visualizador de la viga que permite mostrar distinta información del modelo como la numeración de los nodos, la sección transversal, el material que se utilizó, cargas, escala de los datos utilizados, el caso de carga y datos sobre la renderización del modelo. El segundo complemento en la Figura 54 es el visualizador de vigas, permite mostrar los esfuerzos actuantes en el modelo y también tiene ajustes de renderizado que permiten mostrar los esfuerzos, la utilización de la viga, esfuerzo axial, ver la sección transversal y su desplazamiento, todo esto con distintos colores y formas.

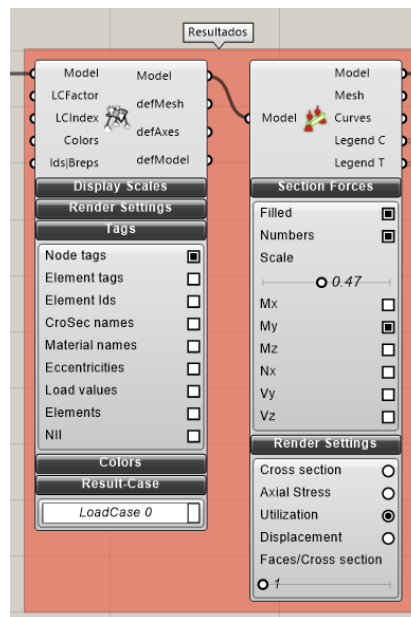


Figura 54. Resultados del Modelo

La Figura 55 contiene el momento actuante en el caso de carga 0 (momento aplicado para la combinación de cargas SLS). En este caso de carga presenta un momento máximo de -52 kN*m, una compresión de 75 kN y un cortante máximo de 13 kN.

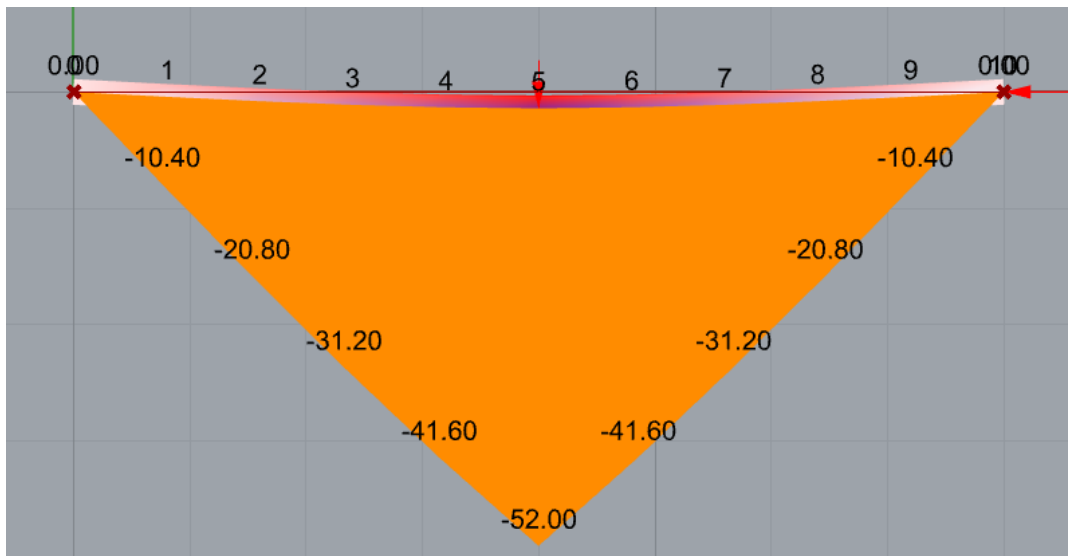


Figura 55. Momento Actuante en Estado Límite de Servicio

El caso de carga 1 muestra el momento actuante (ULS). Presenta un momento máximo de -70.20 kN*m mostrado en la Figura 56, una compresión de 101.25 kN y un cortante máximo 17.55 kN.



Figura 56. Momento Actuante es Estado Límite Último

Los resultados que se obtuvieron del análisis se encuentran resumidos en la Tabla 3, el cual los muestra por su respectivo caso de carga.

Tabla 3. Resumen de Resultados

	Estado Límite de Servició	Estado Límite Último
Caso de Carga	0	1
Momento máx. (kN*m)	-52	-70.20
Axil máx. (kN)	-75	101.25
Cortante máx. (kN)	13	17.55

4.1.9 Optimización del perfil

La optimización se realizó con el componen interno de Karamba 3D mostrado en la Figura 57, este complemento ayuda a optimizar los perfiles de acero realizando las verificaciones que se encuentran en el Eurocódigo 3 parte 1-1 (EN 1993-1-1, 2005).

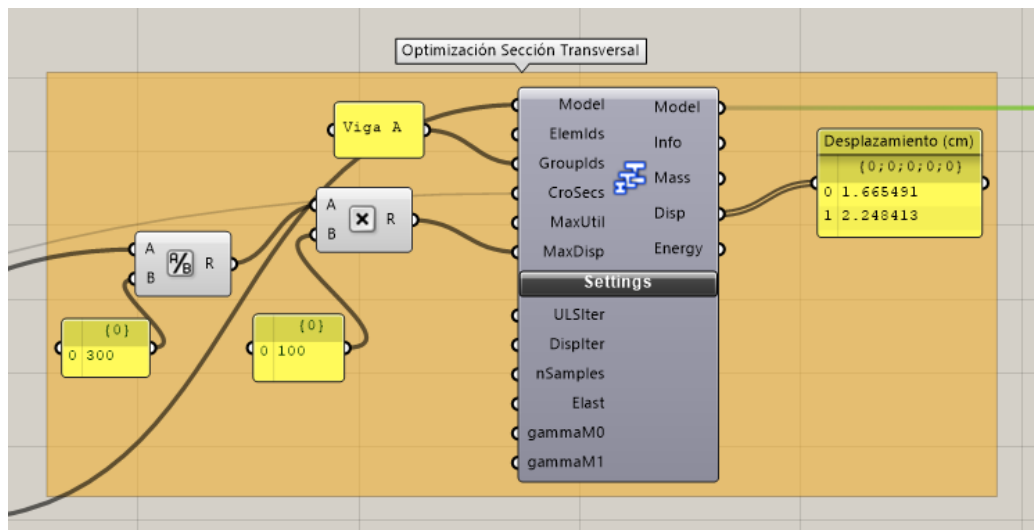


Figura 57. Complemento de Optimización Karamba 3D

Como se mencionó anteriormente este complemento utiliza el Eurocódigo 3 para las verificaciones estructurales, para las revisiones de interacción axil-momento obtiene los factores k_{ij} a través del método 2 del anejo B mostrado en la Figura 58.

Interaction factors	Type of sections	Design assumptions	
		elastic cross-sectional properties class 3, class 4	plastic cross-sectional properties class 1, class 2
k_{yy}	I-sections RHS-sections	$C_{my} \left(1 + 0,6 \bar{\lambda}_y \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{my} \left(1 + 0,6 \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right)$	$C_{my} \left(1 + (\bar{\lambda}_y - 0,2) \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{my} \left(1 + 0,8 \frac{N_{Ed}}{\chi_y N_{Rk} / \gamma_{M1}} \right)$
k_{yz}	I-sections RHS-sections	k_{zz}	$0,6 k_{zz}$
k_{zy}	I-sections RHS-sections	$0,8 k_{yy}$	$0,6 k_{yy}$
k_{zz}	I-sections	$C_{mz} \left(1 + 0,6 \bar{\lambda}_z \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{mz} \left(1 + 0,6 \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$	$C_{mz} \left(1 + (2\bar{\lambda}_z - 0,6) \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{mz} \left(1 + 1,4 \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$
	RHS-sections	$C_{mz} \left(1 + 0,6 \bar{\lambda}_z \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{mz} \left(1 + 0,6 \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$	$C_{mz} \left(1 + (\bar{\lambda}_z - 0,2) \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$ $\leq C_{mz} \left(1 + 0,8 \frac{N_{Ed}}{\chi_z N_{Rk} / \gamma_{M1}} \right)$
For I- and H-sections and rectangular hollow sections under axial compression and uniaxial bending $M_{y,Ed}$ the coefficient k_{zy} may be $k_{zy} = 0$.			

Figura 58. Anejo B Método 2 para Obtener Factores k_{ij} . Citado de (EN 1993-1-1, 2005)

El parámetro gamma M0 y gamma M1 para las verificaciones se toma como 1 pero es modificable.

Uno de los parámetros requeridos para la optimización es la flecha máximo permisible. Para la flecha máxima se tomó como referencia la tabla 3.1 del manual del Eurocódigo 1 en la Figura 59 seleccionando el valor permisible de $L/300$ de acuerdo a normativas.

Condition	δ_{max}	δ_2
(a) Roofs generally	$L/200$	$L/300$
(b) Roofs with access other than for maintenance	$L/250$	$L/300$
(c) Floors generally	$L/250$	$L/300$
(d) Floors/roofs supporting plaster or other brittle finish or non-flexible partition	$L/250$	$L/350$
(e) Floors supporting columns (unless the deflection has been taken into account within global analysis for the ultimate limit state)	$L/400$	$L/500$
(f) When δ_{max} can impair the appearance of the building	$L/250$	—

Figura 59. Valores Límite Recomendados para Flechas Verticales. Citado de (Gulvanessian & Holicky, 1996)

Con los parámetros asignados se obtuvo como resultado la viga HEB 220 en la Figura 60.

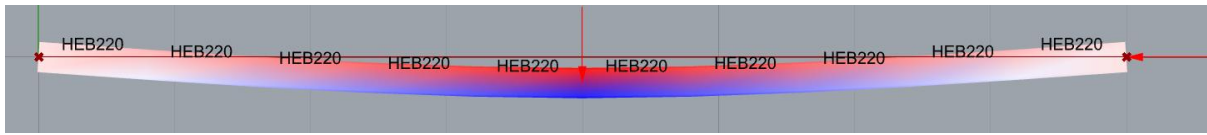


Figura 60. Resultado de Optimización

Un error al utilizar la herramienta de Karamba 3D para optimizar los perfiles es que las verificaciones y las flechas las obtiene del resultado más grande entre los casos asignados como parámetros en las cargas, es decir que la verificación la realiza con las cargas de estado límite último y las flechas también las verifica el resultado obtenido por las cargas en estado límite último sin tomar en cuenta el estado límite de servicio. Probablemente la viga se podría solucionar con un perfil HEB menor, pero al tener una flecha mayor se asignó un perfil más grande. Se puede considerar como que el diseño es seguro ya que el error se encuentra del lado de la seguridad, pero en estructuras complejas esto podría significar mucho dinero.

La Tabla 4 resume los parámetros utilizados en la verificación del componente optimizador de Karamba 3D, también contiene el perfil seleccionado, el ratio de utilización y la flecha obtenida. La flecha obtenida fue de 2.25 cm obtenida del complemento de optimización de Karamba 3D en la Figura 57 y se encuentra muy cercana a la flecha admisible de 2.66 cm, mientras que el ratio de utilización fue de 0.59 que se encuentra lejano del 1, esto quiere decir que el factor predominante durante la optimización para este caso fue el desplazamiento.

La utilización del elemento se obtuvo del complemento llamado utilización de los elementos de Karamba 3D en la Figura 61, este complemento obtiene los ratios de utilización de acuerdo al Eurocódigo 3 (EN 1993-1-1, 2005) y también despliega los valores utilizados en la verificación del componente de optimización de perfiles.

Tabla 4. Resumen de Parámetros Utilizados en Verificaciones de Perfil HEB 220

Perfil Seleccionado	HEB 220
Ratio de Utilización	0.59
Flecha Admisible (cm)	2.67
Flecha Obtenida (cm)	2.25
NRd (kN)	2503.6
VyRd (kN)	1137.44
MtRd (kN*m)	7.62
MyRd (kN*m)	202.26
MzRd (kN*m)	71.09
Mcr (kN*m)	256.39
Ncry (kN)	2620.24
Ncrz (kN)	920.7
CMy	1
CMz	1
CMLT	1
xy	0.6114
xz	0.2708
χLT Mod	0.7669
kyy	1.0388
kzz	1.0896
kyz	1.0896
kzy	0.9836

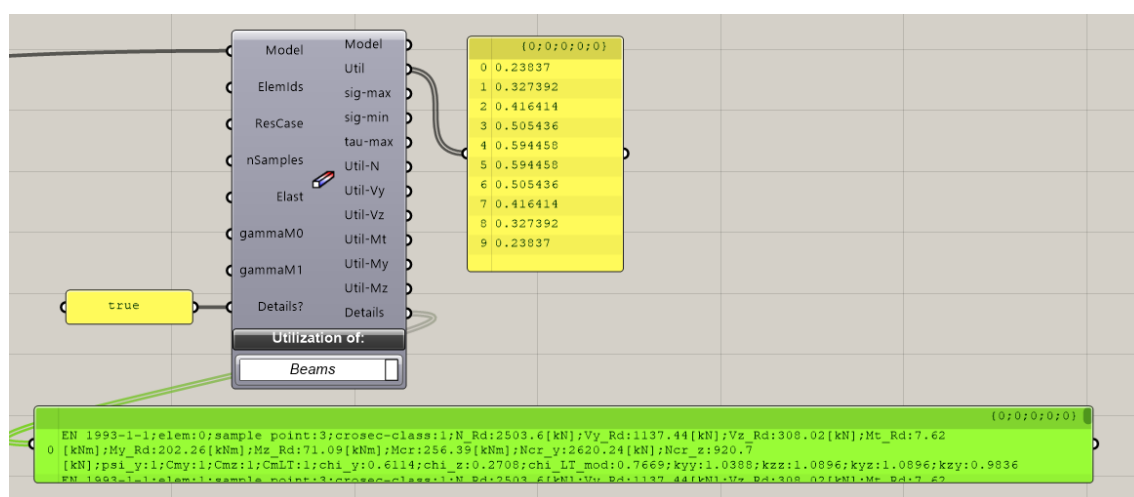


Figura 61. Componente de Utilización de Elementos

Para explorar la herramienta de optimización de Karamba 3D se decidió aumentar el catálogo de opciones para el componente y se agregaron los perfiles IPE como se muestra la Figura 62.

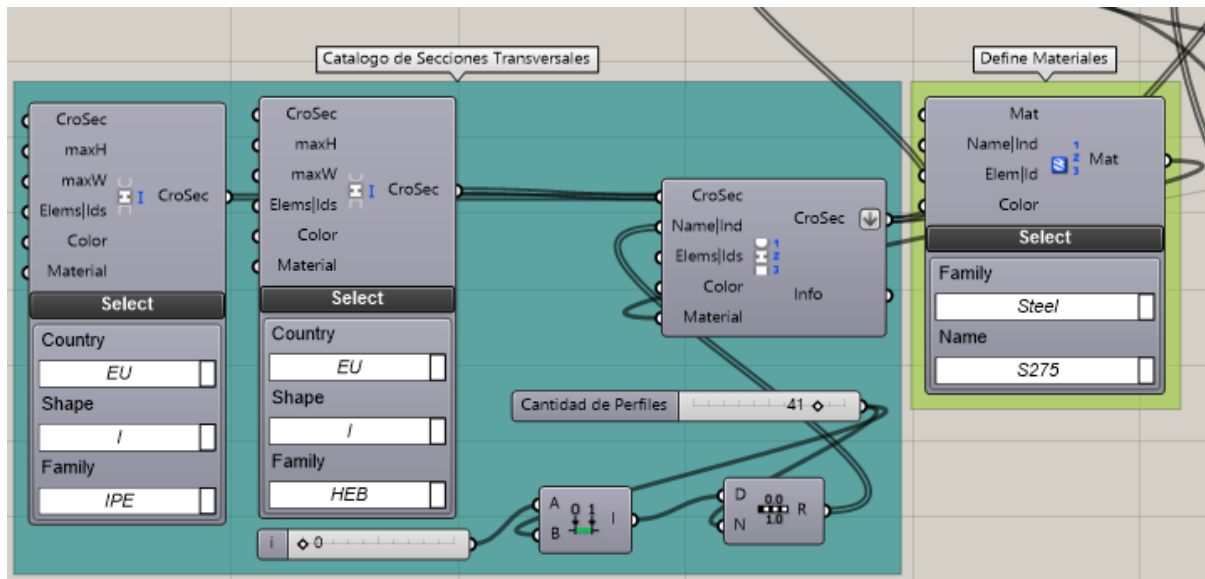


Figura 62. Complemento de Catálogo con Perfiles IPE

Al agregar el nuevo complemento los posibles perfiles abarcan desde HEB 100 hasta HEB 1000 y desde IPE 80 hasta IPE 600. Con el nuevo catálogo el nuevo perfil seleccionado fue el IPE 400 de la Figura 63.

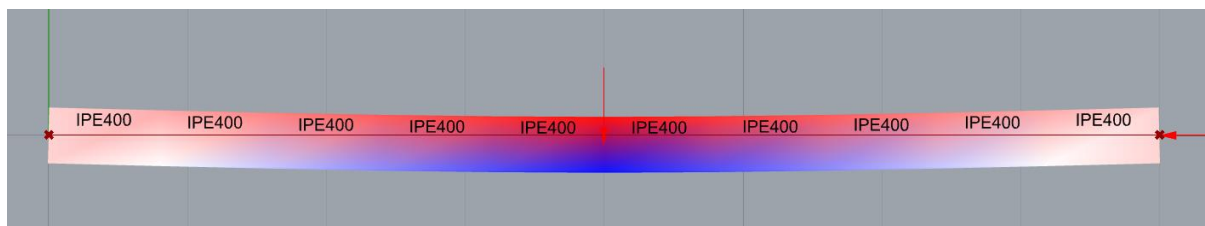


Figura 63. Perfil IPE 400, Obtenido con Nuevo Catálogo

La Tabla 5 resume los parámetros utilizados en la verificación del componente optimizador de Karamba 3D. La Tabla 5 contiene el perfil seleccionado, el ratio de utilización y la flecha obtenida. La flecha fue de 0.79 cm y se encuentra muy lejana a la flecha admisible de 2.66 cm, mientras que el ratio de utilización fue de 0.78 que se encuentra más cercano al 1 que el caso anterior, esto quiere decir que el factor predominante durante la optimización fue el ratio de utilización del elemento. Un perfil HEB 220 pesa 71.5 kg/m, mientras que un perfil IPE 400 tiene un peso de 66.3 kg/m. Con un catálogo mayor de perfiles es posible obtener un resultado óptimo, pero con la limitante de no poder verificar la flecha con el estado de carga deseado, una posible solución sería realizar la optimización verificando la resistencia y estabilidad del elemento estado limite último y de manera separada verificar el cumplimiento de la flecha en

estado límite de servicio, de esta forma evitar la optimización de ambos casos de carga en conjunto y evitar estructuras sobre diseñadas. Un ejemplo donde sería necesario evaluar los resultados con estado límite de servicio y estado límite último por separado sería en un elemento de gran longitud y donde la carga no es muy grande, en este caso sería sencillo que el elemento cumpliera por resistencia, pero en estado límite último la flecha podría variar significativamente con respecto a la de flecha en estado límite de servicio, por lo que sería necesario tener especial cuidado en este tipo de casos al realizar la optimización.

Tabla 5. Resumen de Parámetros Utilizados en Verificaciones de Perfil IPE 400

Perfil Seleccionado	IPE 400
Ratio de Utilización	0.78
Flecha Admisible (cm)	2.67
Flecha Obtenida (cm)	0.79
NRd (kN)	2323.75
VyRd (kN)	814.5
MtRd (kN*m)	6.04
MyRd (kN*m)	317.9
MzRd (kN*m)	40.26
Mcr (kN*m)	156.18
Ncry (kN)	7490.57
Ncrz (kN)	426.83
CMy	1
CMz	1
CMLT	1
χ_y	0.9056
χ_z	0.1586
χ_{LT} Mod	0.4182
k_{yy}	1.0161
k_{zz}	1.1649
k_{yz}	1.1649
k_{zy}	0.9573

4.2 Segundo caso de estudio

Para el segundo caso de estudio se diseñó una celosía empotrada en sus puntos de apoyo, considerando los siguientes parámetros:

- Luz por cubrir: 20 metros.
- Tipo de acero: S235.
- Carga puntual permanente al centro de la celosía adicional al peso propio de 350 kN.
- Catálogo para seleccionar elementos limitado a perfiles HEA.

Los elementos para la programación gráfica que se utilizaron para este caso se encuentran en la Figura 64. Comparado con el primero caso de estudio este caso tiene una mayor cantidad de complementos ya que la celosía se compone por un mayor número de elementos y esto provoca que el proceso de parametrizarlo sea más extenso. Para este caso se realizó la optimización de la celosía utilizando el complemento de Galápagos y Octopus, de esta forma se analizó las ventajas que se tienen al optimizar de esta manera.

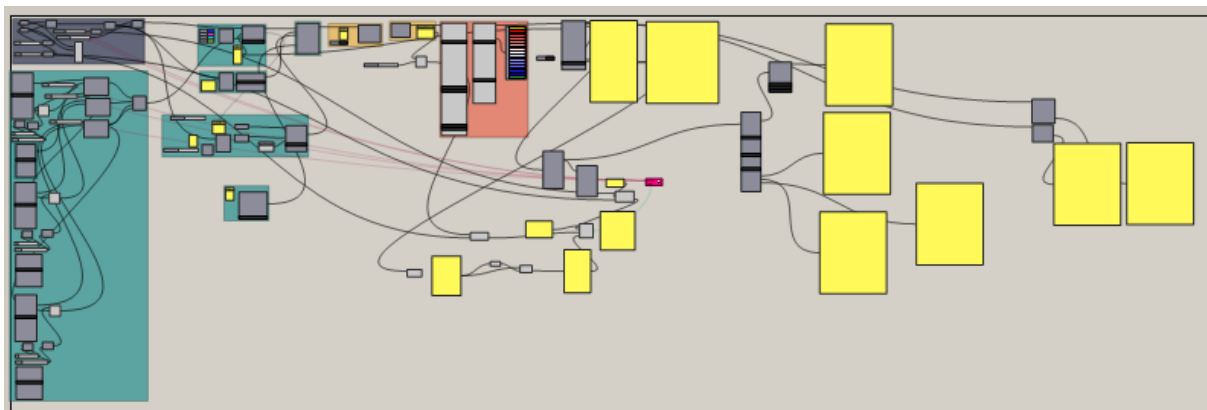


Figura 64. Programación Gráfica en Segundo Caso de Estudio

4.2.1 Criterio para mayoración de cargas

Las cargas permanentes actuantes se mayoraron de acuerdo con la Figura 44. Para estado límite último las cargas se mayoraron por 1.35 mientras que para estado límite de servicio las cargas se mayoraron utilizando el coeficiente de 1.0.

4.2.2 Modelación paramétrica

En la Figura 65 se observa la programación gráfica que se definió en Grasshopper para obtener la geometría de la celosía, la Figura 66 muestra el resultado geométrico obtenido al trasladar el

código al Rhinoceros 3D. La manera en que se definió la geometría fue al trazar una línea entre dos puntos sobre un eje X donde su primer punto se colocó en el origen 0 y el segundo punto se colocó a 6 metros de distancia. Con la línea definida se realiza una extrusión que depende del canto definido en el parámetro, de esta forma generando 2 líneas. Al tener las 2 líneas se divide en el número de divisiones deseado, colocando nodos en las líneas para generar los segmentos requeridos. Teniendo las líneas superior e inferior, así como sus nodos correspondientes se generaron líneas que cruzan entre los nodos como se observa en la Figura 66.

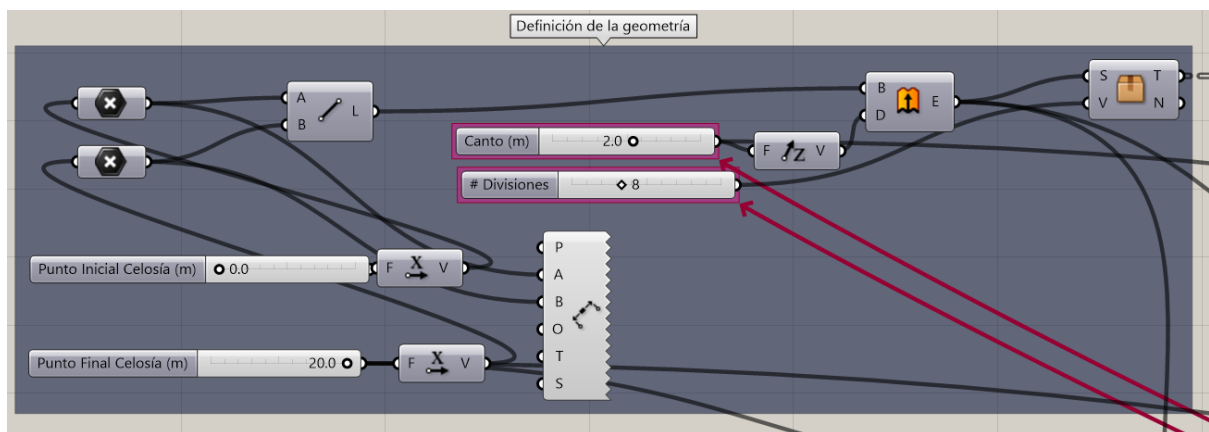


Figura 65. Definición Geométrica de la Celosía Programada

Deslizando los valores en los parámetros de punto inicial y punto final en la Figura 65 es como se puede variar la longitud de la celosía, de igual forma deslizando los parámetros de canto y número de divisiones modifica instantáneamente esos parámetros en la geometría.

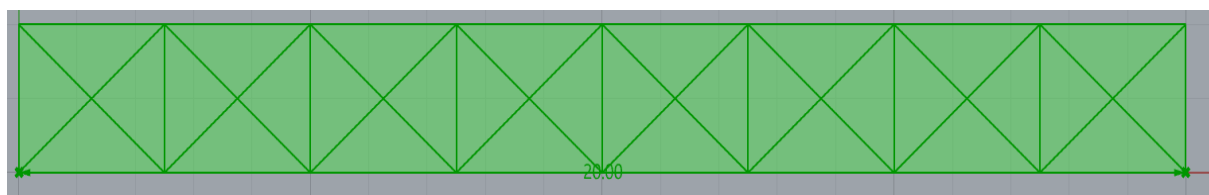


Figura 66. Modelo Visual de la Geometría de la Celosía

4.2.3 Simulación estructural

Para realizar una comunicación entre la geometría programada en Grasshopper y Karamba 3D se utiliza el complemento de la Figura 67 el cual ayuda a definir el elemento como una geometría estructural. A diferencia de la viga del primer caso de estudio en este caso se tienen distintos elementos, estos elementos se deben diferenciar desde este punto del modelo de esta manera se pueden trabajar en procesos posteriores. Los estribos superiores e inferiores se designaron de tipo A, los elementos verticales se clasificaron de tipo B y los elementos diagonales que se encontraban en la celosía desde la mitad hacia el lado izquierdo se denominaron como C2, mientras que las del lado derecho se denominaron como C1. La Figura 68 muestra el modelo de la celosía con sus elementos clasificados.

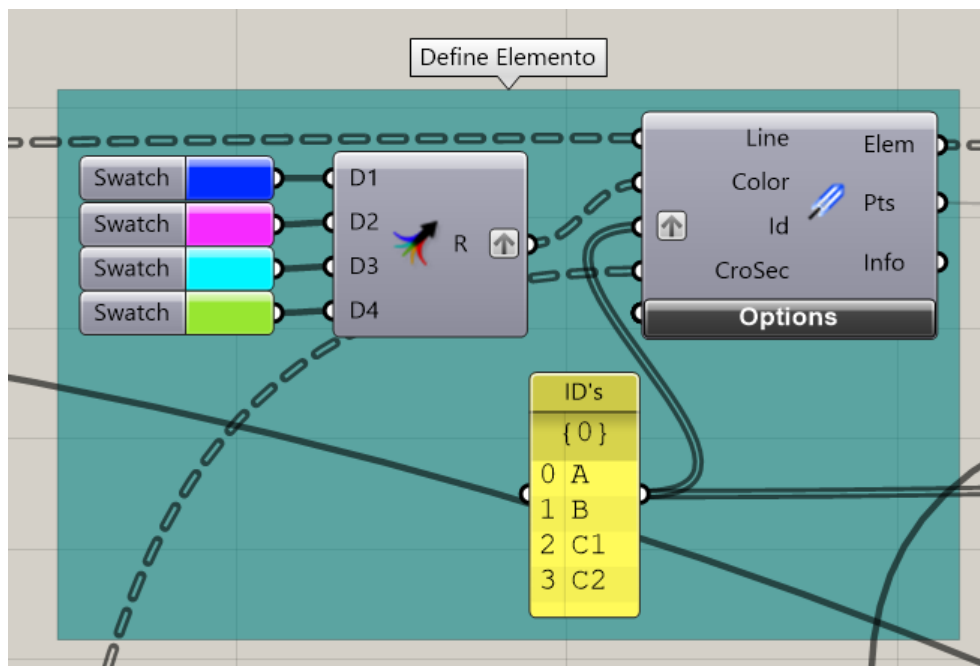


Figura 67. Definición de Geometría como Elemento en Karamba 3D

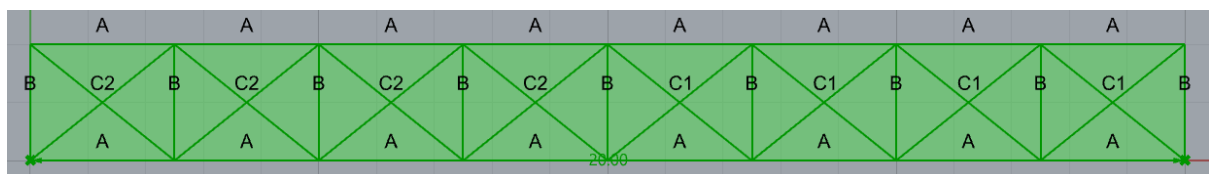


Figura 68. Clasificación de los Elementos de acuerdo con ID's

4.2.4 Definición de los apoyos

Con los nodos y los elementos ya definidos se procedió a colocar los apoyos de tal forma que la celosía se encuentra empotrada en sus extremos. Para el segundo caso de estudio los apoyos se asignaron de forma diferente al primer caso, en este caso se colocaron las coordenadas de la geometría en donde se asignan los puntos como muestra la Figura 69. La geometría se encuentra modelada de tal forma que, si se modifica la longitud o el número de segmentos, las coordenadas de los puntos de apoyo siempre serán las mismas.

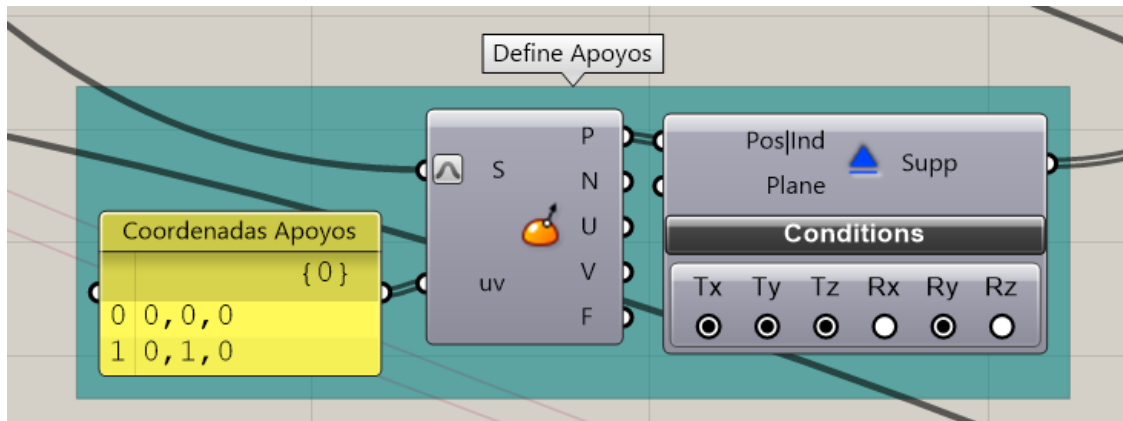


Figura 69. Definición de Apoyos Caso 2

Como parte de los apoyos al ser una celosía se definieron como articulaciones los elementos C1 y C2 mostrado en la Figura 70. Definir las articulaciones en este modelo fue importante ya que si no se definen los elementos diagonales no trabajan en conjunto con la estructura.

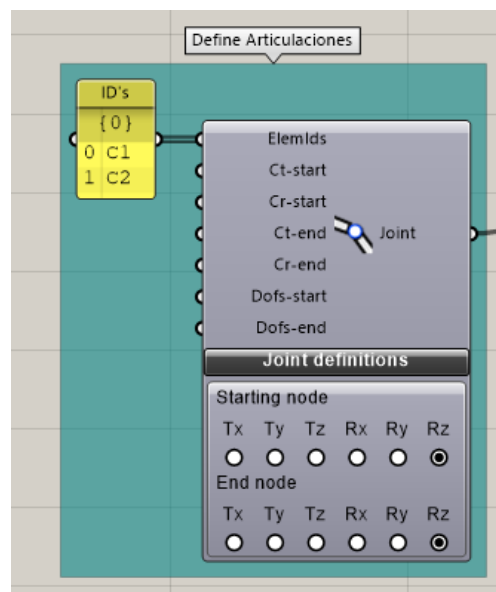


Figura 70. Componente para Definir Articulaciones

4.2.5 Definición de las cargas

La carga se definió de acuerdo con la Figura 71. Esta carga, su posición y dirección se encuentra descrita en la Tabla 6. Al enfrentar la misma limitante que el caso de estudio anterior sobre los casos de carga, se introducen de forma, es decir que se describió el número de caso de carga y se introdujo el valor del caso carga. En la Tabla 7 se describió el caso de carga utilizado para estado límite de servicio y estado límite último, así como sus factores de amplificación.

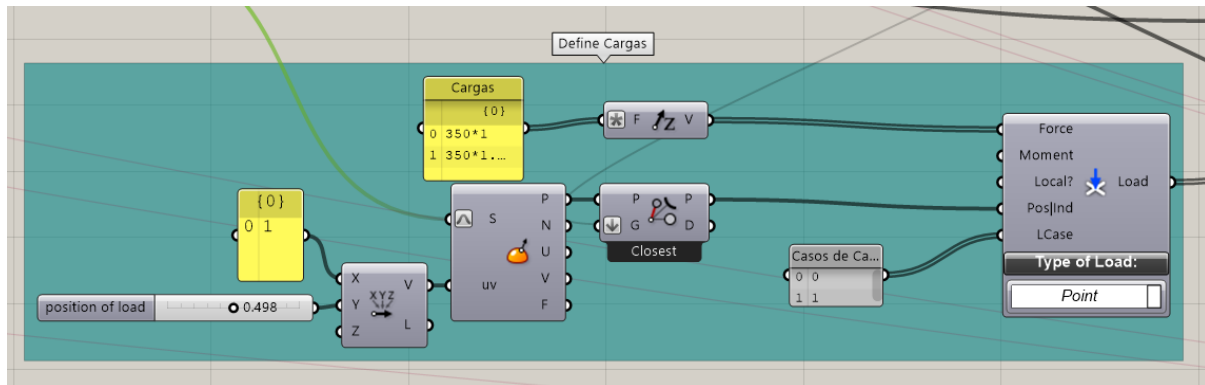


Figura 71. Definición de Carga y Casos de Carga en Karamba 3D

Tabla 6. Magnitud y Posición de Carga

	Nodo Central Superior
Carga (kN)	350
Eje y Dirección	-Z

Tabla 7. Casos de Carga y Factores de Amplificación

	Estado Límite de Servicio	Estado Límite Último
Caso de Carga	0	1
Factor de Amplificación	1	1.35

4.2.6 Definición de materiales

El tipo de material y catálogo de perfiles se programó de acuerdo con la Figura 72.

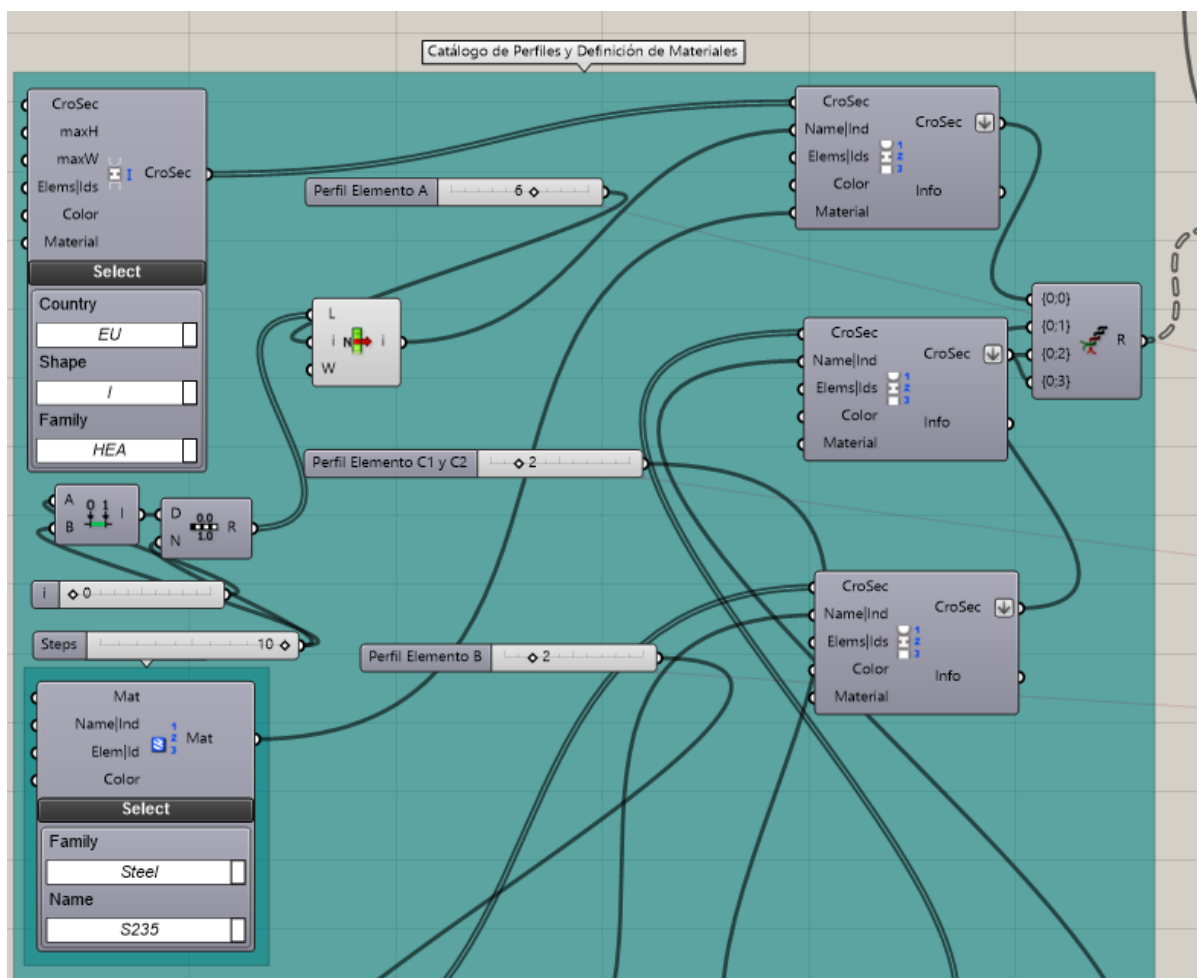


Figura 72. Catálogo de Perfiles y Definición de Materiales

La forma en que funciona es la siguiente, el selector de perfiles por rango mostrado en la Figura 73 provee de un catálogo con perfiles HEA desde el HEA 100 hasta el HEA 1000. El selector de materiales en la Figura 74 determina el tipo de material y su clase, para este caso de estudio se utilizó acero S235. Los componentes deslizantes en la Figura 75 determinan el perfil que se seleccionó de la lista disponible, estos tres componentes eligen un perfil para los elementos tipo A, otro para los tipos B y otro para los elementos tipo C1 y C2, de esta manera se pueden tener diferentes perfiles por tipo de elemento en la celosía. Finalmente, el perfil seleccionado se traslada el selector de perfiles en la Figura 76, este componente contiene la información definitiva sobre el tipo de material y tipo de perfil que se seleccionó para cada elemento.

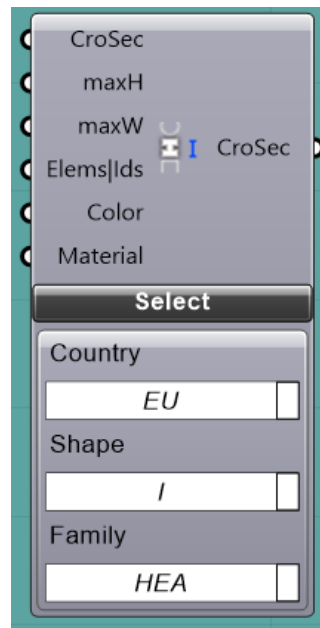


Figura 73. Componente de Selección por Rango de Perfiles

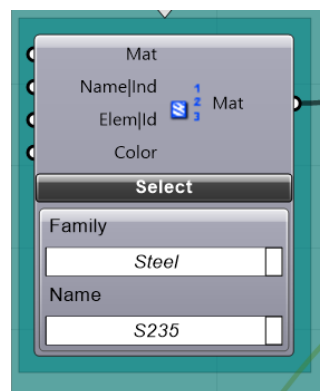


Figura 74. Componente de Selección de Material



Figura 75. Componentes Deslizantes para Seleccionar Perfil

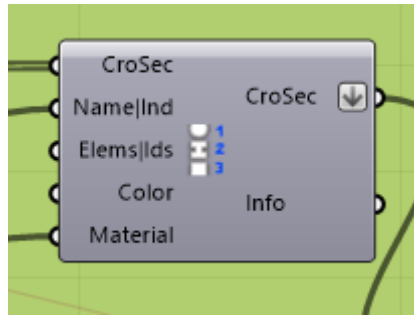


Figura 76. Selector de Perfiles

4.2.7 Ensamblaje del modelo

Con la geometría, los apoyos, la carga y los materiales definidos se ensamblaron en el complemento de Karamba 3D de la Figura 77. Este complemento traduce todo lo que se programó anteriormente para formar un modelo de elementos finitos que se puede manipular en Karamba 3D.

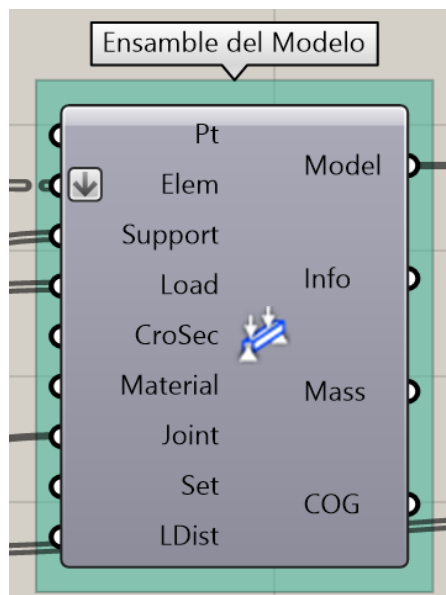


Figura 77. Ensamble del Modelo

4.2.8 Optimización topológica

La celosía contaba con muchas barras por lo que se eliminaron las barras a tensión con el componente de eliminación de elementos en la Figura 78. Los parámetros introducidos fueron los tipos de elementos a eliminar que en este caso fueron los C1 y C2 y el otro parámetro es para indicar que elemento eliminar, para este caso se indicaron los elementos a tensión de los elementos C1 y C2.

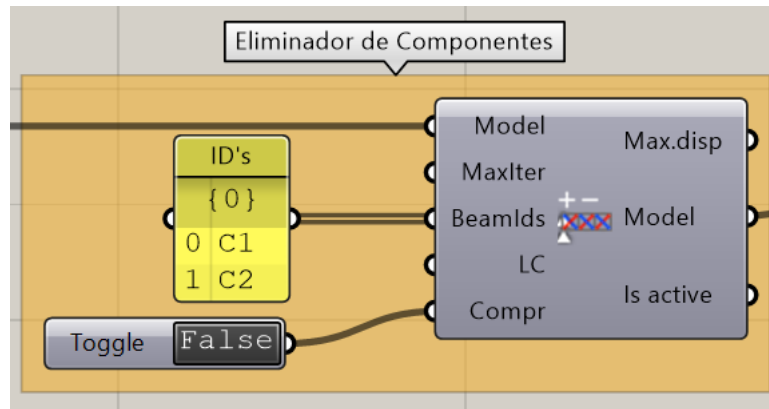


Figura 78. Optimización Topológica Celosía

Al realizar la optimización topológica y de esta manera eliminar los componentes estructurales que no se requieren se obtiene una nueva geometría de la celosía, en la Figura 79 se observa la sucesión de aproximaciones. Realizando esta optimización sobre la geometría reduce en gran cantidad los elementos diagonales de la celosía y por consecuencia el peso.

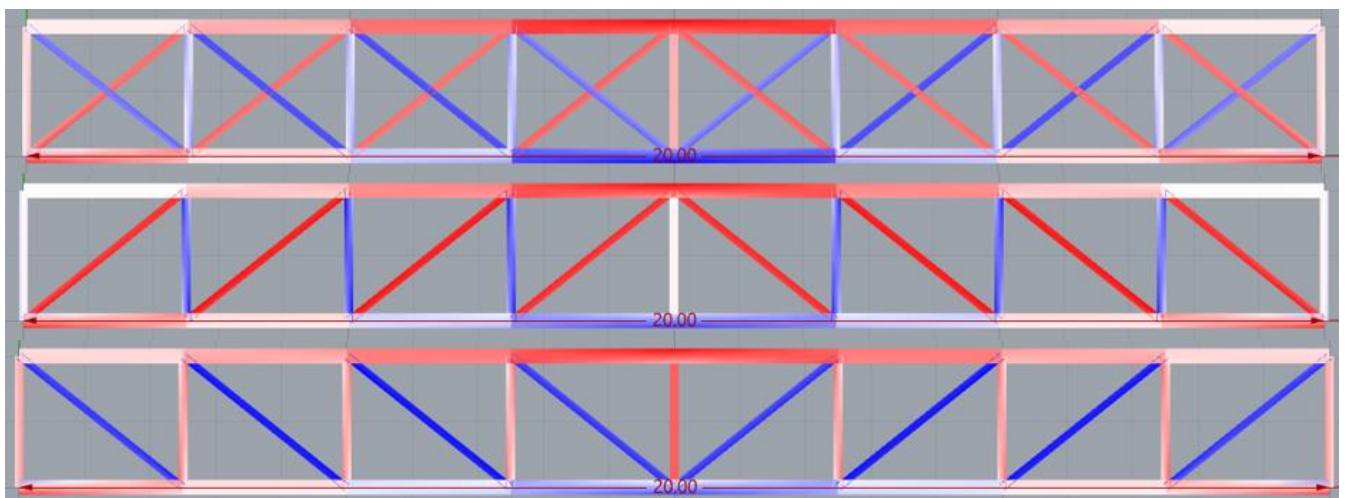


Figura 79. Sucesión de Aproximaciones en Optimización Topológica

4.2.9 Análisis estructural y su resultado

El análisis estructural se realizó con el complemento de análisis en la Figura 80, este complemento obtiene las flechas utilizando teoría de primer orden. El desplazamiento obtenido es para estado límite de servicio se encuentra en el caso de carga 0 y para el estado límite último en el caso de carga 1.

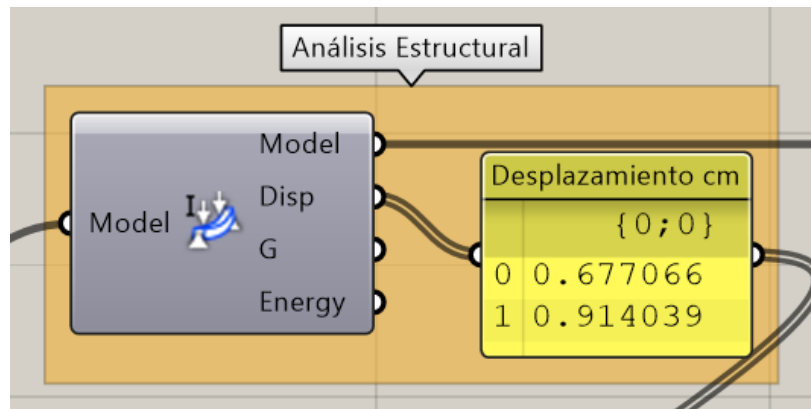


Figura 80. Componente de Análisis Estructural y Desplazamientos

Para observar el modelo y sus resultados se utilizaron los complementos desplegados en la Figura 81. El primero es el visualizador de la viga que permite mostrar distinta información del modelo como la numeración de los nodos, la sección transversal, el material que se utilizó, cargas, escala de los datos utilizados, el caso de carga y datos sobre la renderización del modelo. El segundo complemento en la Figura 81 es el visualizador de vigas, permite mostrar los esfuerzos actuantes en el modelo y también tiene ajustes de renderizado que permiten mostrar los esfuerzos, la utilización de la viga, esfuerzo axial, ver la sección transversal y su desplazamiento, todo esto con distintos colores y formas.

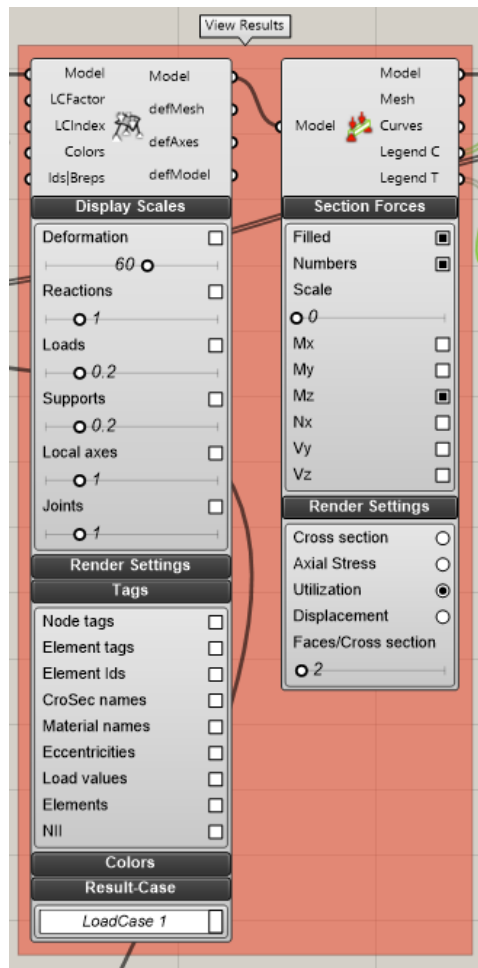


Figura 81. Resultados del Modelo

Con el visor de resultados de la Figura 81 se puede observar la numeración de los elementos de la celosía en la Figura 82. No se puede desplegar gráficamente los esfuerzos actuantes, pero con el componente de esfuerzos resultados mostrados en la Figura 83 se puede desplegar el momento, axil y cortante actuantes es la estructura, dichos resultados se encuentran en la Tabla 8.

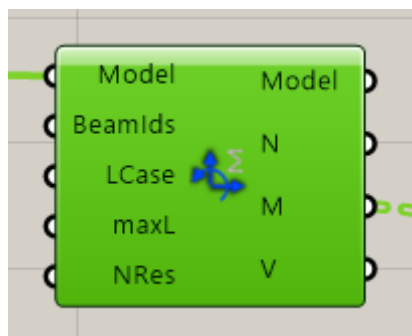


Figura 82. Componente de Esfuerzos Resultantes en la Estructura

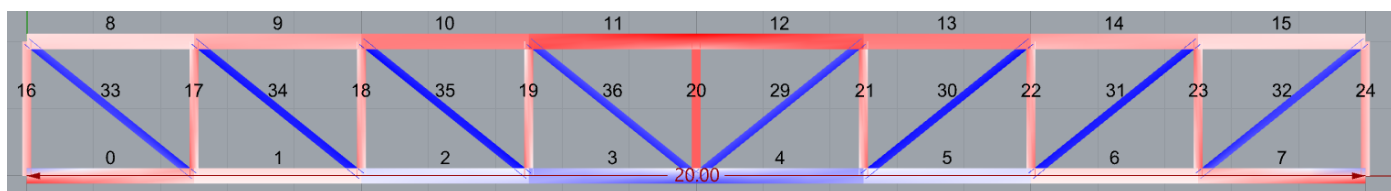


Figura 83. Numeración de los Elementos en la Celosía

Tabla 8. Resumen de Resultados

Elemento	Estado Límite de Servicio			Estado Límite Último		
	Momento máx. (kN*m)	Axil máx. (kN)	Cortante máx. (kN)	Momento máx. (kN*m)	Axil máx. (kN)	Cortante máx. (kN)
0	59.08	-306.49	29.76	79.76	-413.79	40.19
1	5.11	-117.78	0.25	6.90	-159.00	0.34
2	6.83	101.18	3.77	9.22	136.59	5.10
3	32.56	323.09	14.06	43.95	436.17	18.99
4	32.56	323.09	14.06	43.95	436.17	18.99
5	6.83	101.18	3.77	9.22	136.59	5.10
6	5.11	-117.78	0.25	6.90	-159.00	0.34
7	59.08	-306.49	29.77	79.76	-413.76	40.18
8	3.40	-179.29	2.31	4.58	-242.04	3.12
9	8.36	-399.57	5.87	11.28	-539.42	7.93
10	3.61	-619.96	1.50	4.88	-836.95	1.98
11	37.50	-807.98	17.45	50.63	-1090.78	23.56
12	37.50	-807.98	17.45	50.63	-1090.78	23.56
13	3.61	-619.96	1.47	4.88	-836.95	1.98
14	8.36	-399.57	5.87	11.28	-539.42	7.92
15	3.40	-179.29	2.31E+00	4.58	-242.04	3.12E+00
16	4.01	-145.23	1.02	5.42	-196.06	1.38
17	8.96	-172.43	8.39	12.09	-232.78	11.33
18	7.13	-165.35	7.07	9.63	-223.22	9.54
19	8.70	-159.46	8.58	11.75	-215.27	11.59
20	2.61E-09	-315.09	2.57E-09	3.52E-09	-425.37	3.47E-09
21	8.70	-159.46	8.58	11.75	-215.27	11.59
22	7.13	-165.35	7.07	9.63	-223.22	9.54
23	8.96	-172.43	8.39	12.09	-232.78	11.33
24	4.01	-145.23	1.02	5.42	-196.06	1.38
29	1.04	229.74	0.05	1.4	310.15	0.07
30	1.28	272.62	0.71	1.73	368.04	0.95
31	0.88	270.95	0.50	1.19	365.78	0.67
32	1.90	230.08	1.04	2.57	310.61	1.41
33	1.90	230.08	1.04	2.57	310.61	1.41
34	0.88	270.95	0.50	1.19	365.78	0.67
35	1.28	272.62	0.71	1.73	368.04	0.95
36	1.04	229.74	0.05	1.41	310.15	0.07

4.2.10 Optimización estructural

La optimización estructural se realizó en dos fases, primero con el componente de Galápagos y posteriormente con el componente de Galápagos combinado de Octopus para verificar si el resultado obtiene alguna mejora.

4.2.11 Optimización a partir de Galápagos

La optimización con Galápagos se divide en dos fases, optimizar la geometría y optimizar el tamaño de la estructura. Optimizar la geometría se refiere al hecho de variar el peralte y número de segmentos de la celosía, mientras que optimizar el tamaño se refiere a el tipo de perfiles a asignar a cada elemento. El componente de Galápagos se observa en la Figura 84.

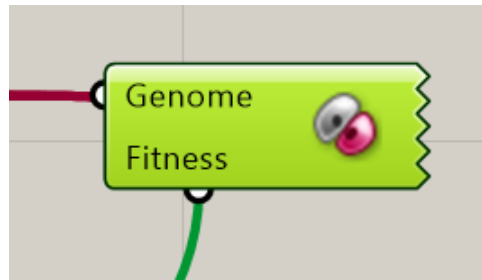


Figura 84. Componente Galápagos

Galápagos cuenta con dos parámetros el “Genome” que es lo que se busca modificar y el “Fitness” que el objetivo que se busca lograr modificando los parámetros en el “Genome”. Galápagos solo puede tener un objetivo para optimizar, es decir que si se quiere una estructura óptima el objetivo del componente será minimizar el peso de la estructura, un desplazamiento que cumpla los requerimientos de diseño y elementos estructurales trabajando lo más cercano posible al 100% de su capacidad. Teniendo en cuenta los requisitos del objetivo a cumplir se utilizó un componente de función observado en la Figura 85. La función establecida se decidió por conveniencia y necesidad de esta manera para este caso de estudio, pero es importante mencionar que se puede personalizar dependiendo de los objetivos que se desean conseguir en el proceso de optimización.

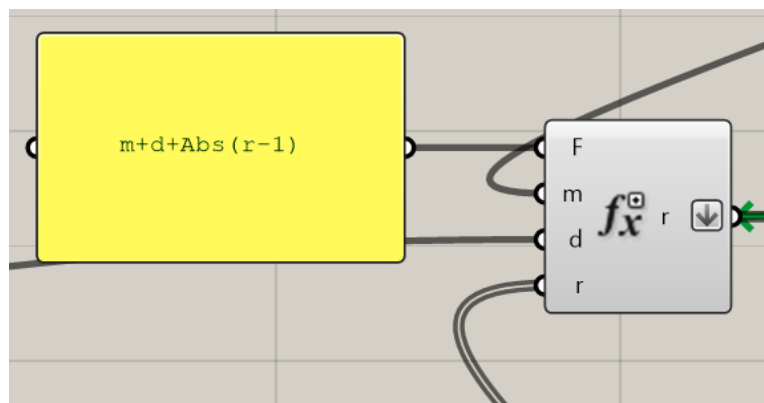


Figura 85. Componente de Función para Optimizar

El componente de la Figura 85 tiene cuatro parámetros de entrada, el primero es la función y los otros tres parámetros son las variables que se utilizan en la función establecida. La función es la suma de la masa de la estructura en el parámetro “m”, la flecha en el parámetro “d” y el valor absoluto de la diferencia entre los ratios de utilización menos uno con la variable “r” para obtener un valor menor que 1. Lo que se buscó es que tomara los ratios de utilización de cada uno de los elementos de la celosía para ir verificando la función con cada uno de los ratios y de esta forma se convirtió el resultado de la función en el parámetro “Fitness” del componente Galápagos. Como se mencionó en el apartado anterior la función objetivo no siempre tiene que ser igual, esta se puede ajustar dependiendo de las necesidades de cada usuario y colocar distintos parámetros. Los parámetros para modificar, es decir el “Genome” para lograr el objetivo se encuentran en la Figura 86.



Figura 86. "Genome" del Componente Galápagos

Las variaciones de los parámetros pueden ser infinitas por lo que se requieren imponer ciertas limitaciones antes de utilizar los parámetros como datos de entrada para la función “Fitness” de Galápagos. En la Figura 87 se encuentra el componente de Visual Basic llamado “limitar Canto” (Ver Anejo 1) este componente se maneja como cualquier otro componente de Grasshopper pero la diferencia es que al acceder dando doble click, se puede programar en código como se observa en la Figura 88 para generar componentes con nuevas funciones personalizadas dentro de Grasshopper, las variables a utilizar en el código se pueden obtener de forma externa, es decir que los valores se pueden conectar al componente de Visual Basic

para utilizarlos como variable o se pueden definir internamente dentro de la ventana de código de Visual Basic. El componente para limitar el canto tiene como parámetros de entrada el canto de la celosía, el peso de la estructura y la longitud, lo que hace es verificar si el canto se encuentra entre $L/15$ y $L/10$, si cumple con el límite el peso de la estructura se mantiene real y si no cumple el peso lo modifica a 10000, de esta forma al generar un número muy grande Galápagos va a cambiar los parámetros hasta cumplir el límite y generar un peso menor. El resultado de este componente se traslada el parámetro “m” en la Figura 85.

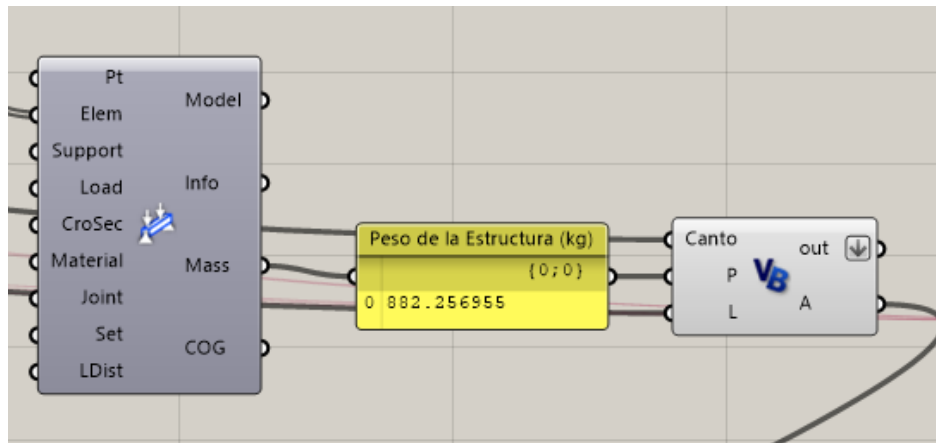


Figura 87. Componente en Código para Limitar Canto

```

Script Editor
Script component: Filtro 1

1 Option Strict Off
2 Option Explicit On
3
4 Imports
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 Public Class Script_Instance
22 Inherits GH_ScriptInstance
23
24 Utility functions
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67 Private Sub RunScript(ByVal x As Double, ByVal y As Object, ByRef A As Object)
68 'convierte a 1000 los valores mayores a 1
69 If x > 1 Then
70 x = 1000
71 End If
72 print(x)
73 End Sub
74
75 '<Custom additional code>
76
77 '</Custom additional code>
78 End Class

```

Figura 88. Venta Para Programar en Código del Componente de Visual Basic

El siguiente parámetro que se filtró fue el desplazamiento con el componente en la Figura 89 llamado “limitar desplazamiento” que fue programado (Ver Anejo 2). En el primer caso de estudio durante la optimización era imposible verificar la flecha en estado límite de servicio y verificar la estructura en estado límite último, con este método es posible realizar la verificación de forma correcta, lo que se hizo fue alimentar el componente “limitar desplazamiento” con el desplazamiento obtenido en estado límite de servicio, de esta manera la optimización se realizó en base al parámetro indicado. Lo que se realizó en este componente fue verificar si el desplazamiento en estado límite de servicio era menor al límite impuesto de $L/300$, este límite se utilizó también durante el primer caso de estudio, si el desplazamiento cumplía se mantenía igual y en caso de no cumplir se aumentaba a 1000, este dato se insertó como parámetro de entrada en la función “Fitness” de Galápagos de esta forma si Galápagos detectaba un valor de desplazamiento muy alto su objetivo iba a ser reducirlo.

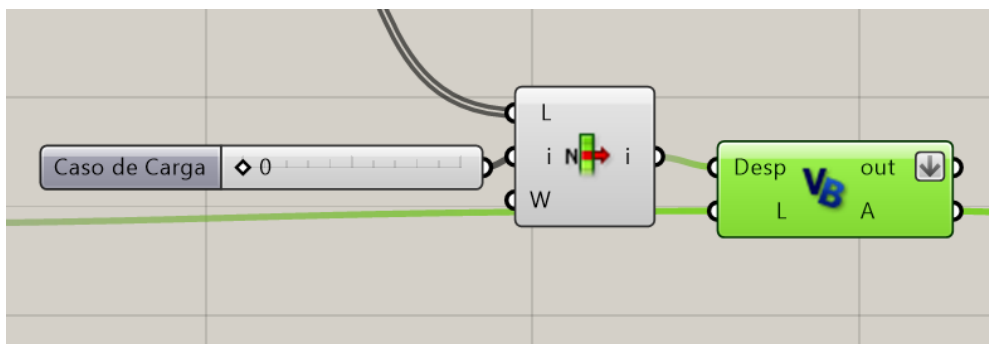


Figura 89. Componente en Código para Limitar Flecha

El siguiente parámetro a filtrar fue los ratios de utilización, estos datos se trataron en diferentes fases. Primero se obtuvieron los ratios con el componente de utilización de la Figura 90, este componente obtiene los ratios de acuerdo al Eurocódigo 3 y el Anexo B mostrado en la Figura 58, también numera los ratios de acuerdo a la numeración de los elementos de la Figura 83. Los ratios mostrados se obtienen a partir del caso de carga que genera mayores esfuerzos, es decir que los obtiene a partir de estado límite último.

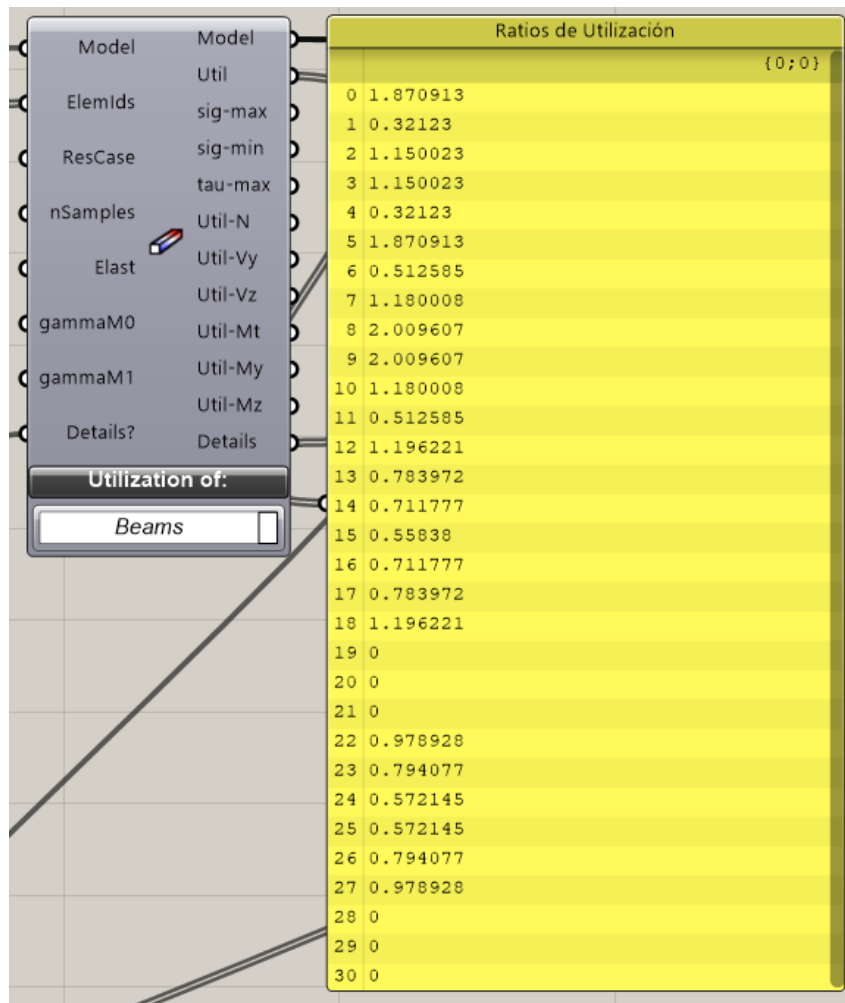


Figura 90. Componente de Utilización

Una vez que se obtuvieron los ratios se utiliza el componente programado llamado “verifica ratio” de la Figura 91 (Ver Anejo 3), este componente tiene como datos de entrada los ratios de utilización, filtra los valores de la lista obtenida y verifica que ningún valor sea mayor a 1, ya que si alguno es mayor a 1 el elemento en cuestión va a fallar. Si uno de los valores es mayor a 1 lo modifica a 1000.

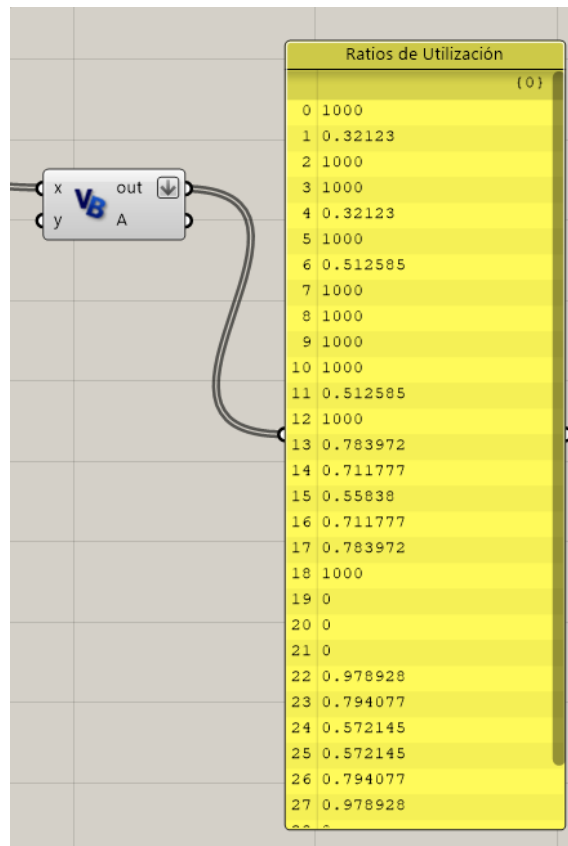


Figura 91. Componente en Código para Verificar Ratios

Al terminar el filtro de información los datos pasan por otro componente programado llamado “convertir a mil” de la Figura 92 (Ver Anejo 4), este componente tiene como datos de entrada la lista de ratios modificada de la Figura 91 y un componente que mide el tamaño de la lista introducida, lo que hace es que si detecta un valor de 1000 en la lista convierte todos los valores a 1000, esto se hace porque la lista de ratios se conecta como parámetro de entrada en la función de optimización de Galápagos y si detecta algunos números en 1000 y otros más pequeños se va a confundir y no va a optimizar de manera adecuada teniendo como resultado ratios de utilización mayores a 1. La lista obtenida se introduce como parámetro “r” en la función de optimización de la Figura 85. Cabe destacar que la estrategia utilizada para colocar las limitaciones y los filtros necesarios para la optimización se deshicieron de manera personal para este caso de estudio, se pueden obtener los mismos resultados realizando diferentes procedimientos y siguiendo distintas estrategias.

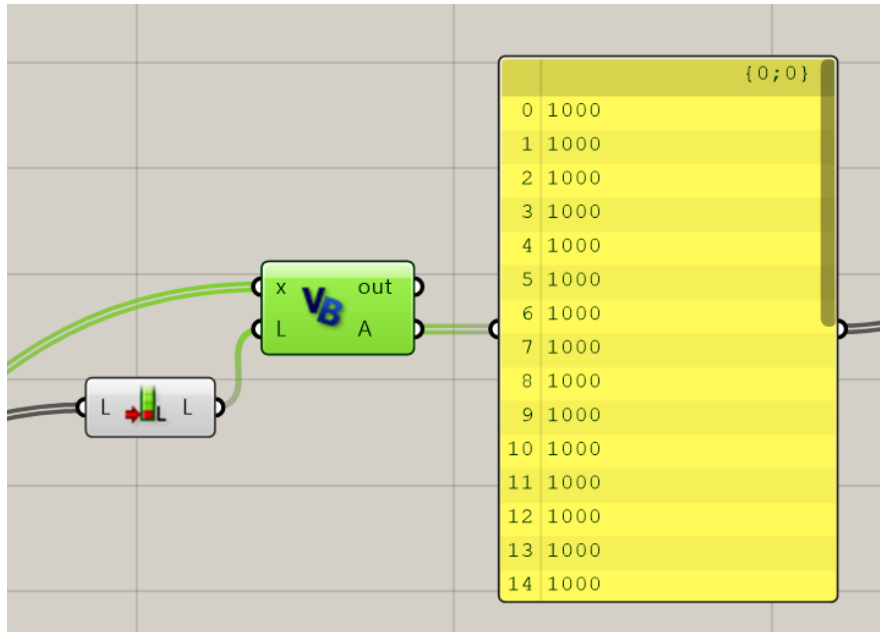


Figura 92. Componente en Código para Convertir a Mil

Al tratar los datos se obtiene una lista de resultados mostrado en la Figura 93, esta lista suma el peso de la estructura, la flecha de la estructura y el ratio de cada elemento de la celosía, este resultado se vuelve el objetivo de Galápagos y se busca minimizar estos valores para obtener el resultado óptimo.

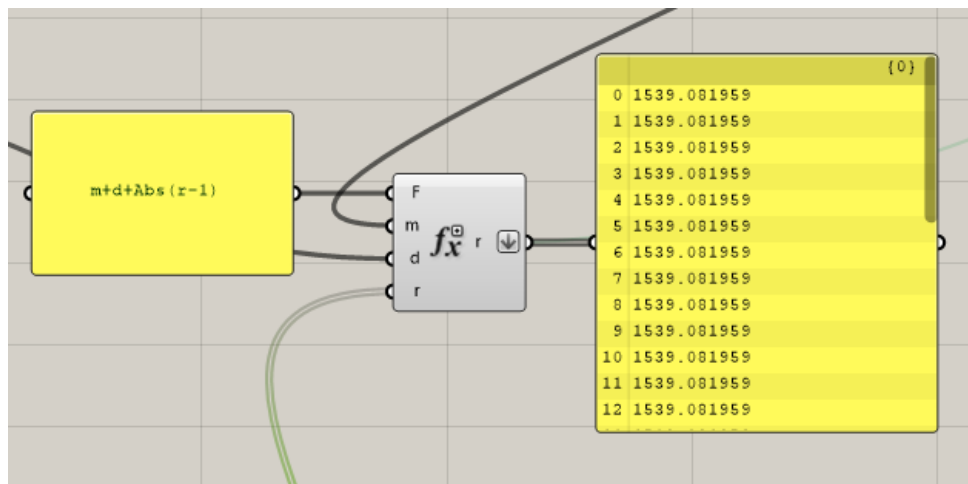


Figura 93. Datos Objetivo de Galápagos

Con la función lista se ejecutó el componente de Galápagos mostrado en la Figura 94, este comienza a hacer simulaciones variando los parámetros que se establecieron, cumpliendo las restricciones que se impusieron y tomando en cuenta distintas posibilidades en las combinaciones de los parámetros.

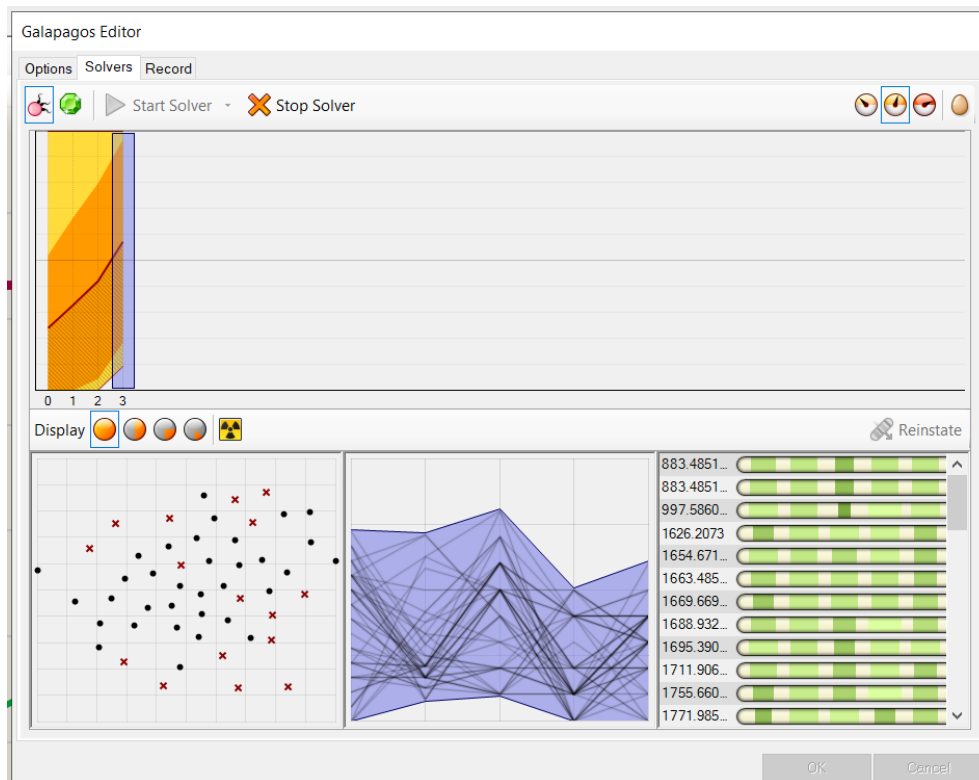


Figura 94. Componente Galápagos Durante Simulación

El resultado obtenido por Galápagos se muestra en la Figura 95. Los perfiles que se seleccionaron se encuentran en la Tabla 9 y el canto, peso y número de divisiones que resultaron se encuentran en la Tabla 10.

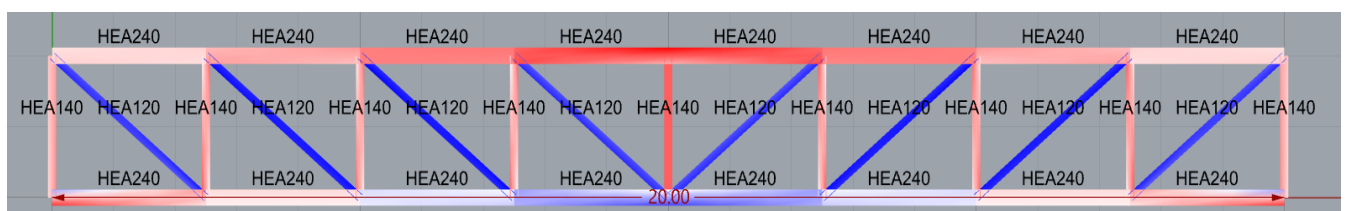


Figura 95. Estructura Obtenida con Galápagos

Tabla 9. Perfiles Obtenidos con Galápagos

Elemento	Perfil
A	HEA240
B	HEA140
C1	HEA120
C2	HEA120

Tabla 10. Parámetros de la Celosía Obtenidos con Galápagos

Resultados	
Canto (m)	2
# de Divisiones	8
Peso (kg)	3366.22
Flecha (cm)	2.63

Los ratios de utilización obtenidos con Galápagos se muestran en la Figura 96, dichos ratios se encuentran numerados de acuerdo con la Figura 83. Muchos de los valores se encuentran cercanos a 1 y algunos no tanto, eso se debe a que al ser una carga puntual los elementos que más trabajan se encuentran cerca de la carga, los elementos con menor ratio de utilización se encuentran más alejados, pero al encontrarse vinculados por el mismo tipo de elemento todos los elementos tipo A son iguales, todos los tipo B son iguales y los C1 y C2 son iguales, si se quisiera tener una mayor optimización se podría permitir variabilidad entre los elementos internos de la celosía y aumentar el catálogo de perfiles para tener mayores posibilidades de ajustar un perfil más adecuado. Es importante mencionar que los valores marcados como 0 en la Figura 96 se debe a que el elemento con ese número no existe en el modelo.

Ratios de Utilización	
	{0;0}
0	0.754779
1	0.143913
2	0.133777
3	0.518588
4	0.518588
5	0.133777
6	0.143913
7	0.754779
8	0.181912
9	0.411017
10	0.559774
11	0.993252
12	0.993252
13	0.559774
14	0.411017
15	0.181912
16	0.483931
17	0.7238
18	0.642277
19	0.685212
20	0.736312
21	0.685212
22	0.642277
23	0.7238
24	0.483931
25	0
26	0
27	0
28	0
29	0.583383
30	0.687349
31	0.662014
32	0.624571
33	0.624571
34	0.662014
35	0.687349
36	0.583383

Figura 96. Ratios de Utilización Obtenidos por Galápagos

4.2.12 Optimización a partir de Octopus y Galápagos

A diferencia de Galápagos, Octopus tiene la opción de cumplir objetivos múltiples, es decir que, en lugar de hacer una ecuación para obtener un valor a cumplir, a Octopus se le puede asignar múltiples valores como objetivo de optimización. La desventaja de Octopus es que no puede leer la lista de ratios de utilización como objetivo y es por esto que se debe hacer una mezcla con Galápagos.

Manteniendo las mismas limitaciones impuestas en los datos mostrando en la Figura 87, Figura 89, Figura 91 y Figura 92 se establecen los objetivos para Octopus y Galápagos. Los parámetros a modificar para Octopus son el canto y el número de segmentos y los objetivos a verificar son el peso y la flecha, es decir que se hizo una optimización de la geometría. Al ejecutar el componente se abre una ventana que comienza a realizar simulaciones mostrada en la Figura 97, en esta ventana se generan distintos cubos que representan cada una de las diferentes

soluciones obtenidas durante el proceso de optimización, no todos son buenos resultados, simplemente muestra los resultados que se van obteniendo durante sus iteraciones y dependiendo en que espacio de la gráfica los ubica los parámetros son diferentes, por ejemplo si la estructura se encuentra en un nivel más alto del eje y sus segmentos son más grandes o su tamaño es más grande.

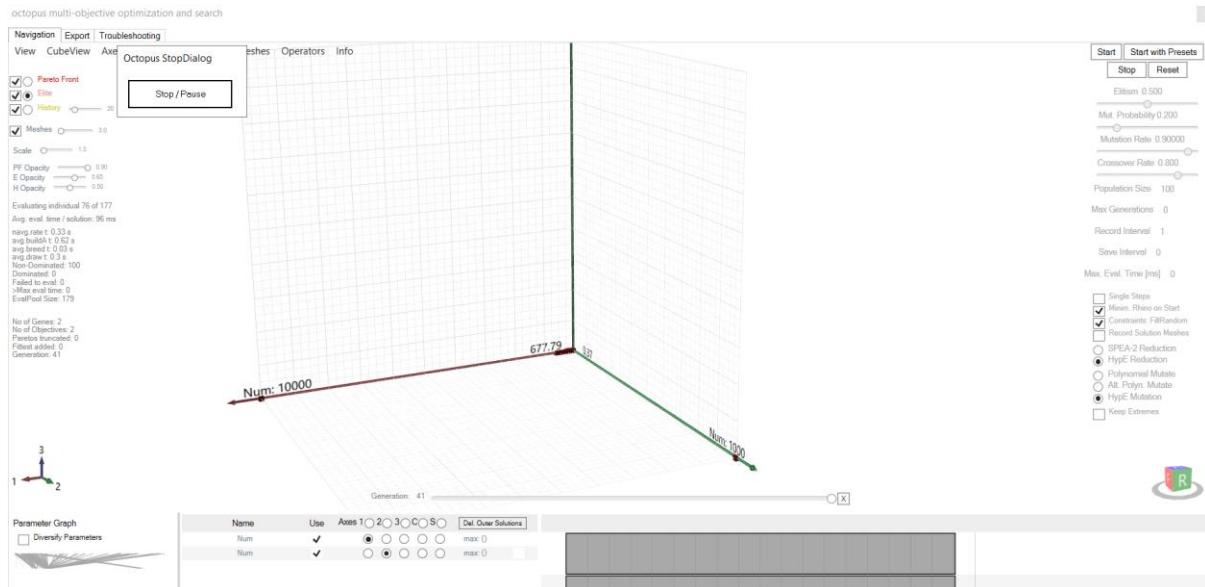


Figura 97. Optimización Geométrica Octopus

Una vez que finalizó el simulador de Octopus se utilizó el componente de Galápagos. Los parámetros a modificar para Galápagos fueron los perfiles de los elementos de la celosía y su objetivo a verificar fueron los ratios de utilización, como no se puede optimizar directamente con los ratios se utilizó la ecuación de la Figura 98, esta pequeña función renta a 1 el valor de los ratios y esto hace que al minimizar el resultado de esta función los ratios reales sean lo más cercanos posibles a 1. La optimización obtenida con Galápagos es la optimización de tamaño de los perfiles. Se requirió repetir el proceso de utilizar Octopus y Galápagos ya que al utilizar dos componentes que ajustan diferentes parámetros de la estructura se debe realizar un proceso iterativo hasta que el resultado converja, este proceso se realizó tres veces.

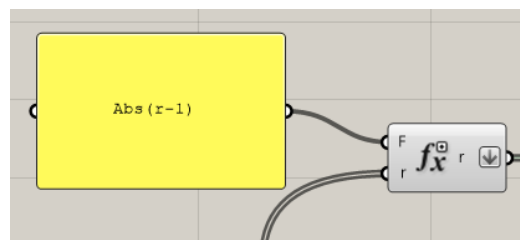


Figura 98. Función para Optimizar Ratios de Utilización en Galápagos

El modelo que resultó de la primera iteración se muestra en la Figura 99. La Tabla 11 contiene un resumen de los perfiles asignados a la celosía y la Tabla 12 muestra el canto, número de segmentos, peso y flechas.

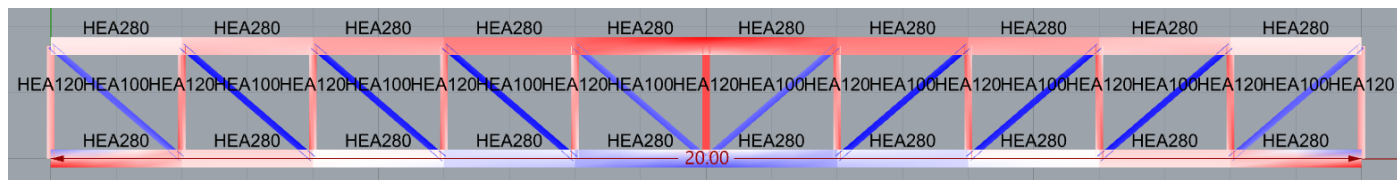


Figura 99. Modelo Primera Iteración con Octopus y Galápagos

Tabla 11. Perfiles Seleccionados para Celosía en Primera Iteración

Elemento	Perfil
A	HEA280
B	HEA120
C1	HEA100
C2	HEA100

Tabla 12. Parámetros de Celosía Obtenidos en Primera Iteración

Resultados	
Canto (m)	1.7
# de Divisiones	10
Peso (kg)	3761.57
Flecha (cm)	2.98

La Figura 100 muestra el resultado para la segunda iteración y la tercera ya que convergió durante la tercera iteración. La Tabla 13. Perfiles Seleccionados para Celosía en Segunda y Tercera Iteración contiene un resumen de los perfiles asignados a la celosía y la Tabla 14 muestra el canto, número de segmentos, peso y flechas. Los elementos numerados se encuentran muestran en la Figura 101.

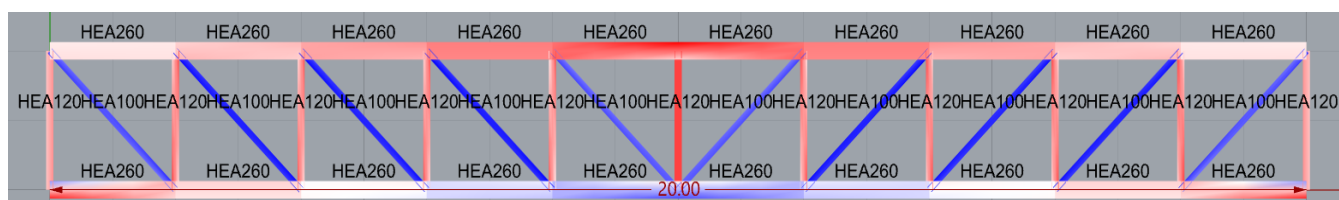


Figura 100. Modelo Segunda y Tercera Iteración con Octopus y Galápagos

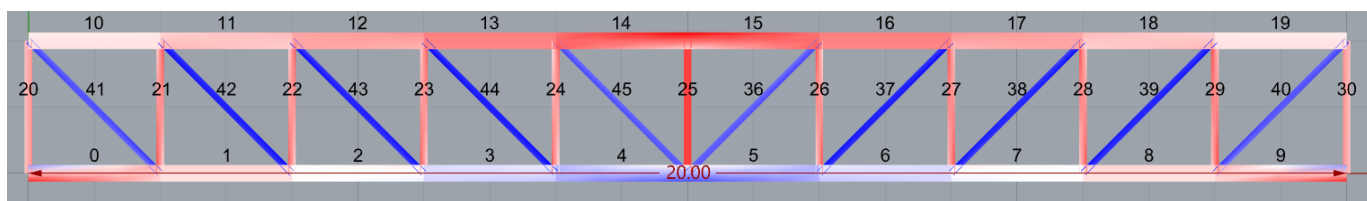


Figura 101. Numeración de Elementos Segunda y Tercera Iteración con Octopus y Galápagos

Tabla 13. Perfiles Seleccionados para Celosía en Segunda y Tercera Iteración

Elemento	Perfil
A	HEA260
B	HEA120
C1	HEA100
C2	HEA100

Tabla 14. Parámetros de Celosía Obtenidos en Segunda y Tercera Iteración

Resultados	
Canto (m)	2
# de Divisiones	10
Peso (kg)	3635.36
Flecha (cm)	2.51

La Tabla 15 contiene una comparación de resultados entre ambos métodos realizados, tanto con el componente Galápagos y el método utilizando Octopus junto con Galápagos. El segundo método tiene sus elementos con mayores ratios de utilización, pero en cuanto a peso, flecha y una menor diferencia entre el tamaño de los elementos utilizados el primero método consiguió mejores resultados. Los ratios de utilización en la Tabla 15 se encuentran numerados de acuerdo a la Figura 83 y Figura 101.

Tabla 15. Comparación de Resultados entre Galápagos y Octopus en Conjunto con Galápagos

Comparación de Resultados entre los Métodos		
Método de Optimización	Galápagos	Octopus y Galápagos
Canto (m)	2	2
# de Divisiones	8	10
Peso (kg)	3366.22	3635.36
Flecha (cm)	2.63	2.51
Perfil Elemento A	HEA240	HEA260
Perfil Elemento B	HEA140	HEA120
Perfil Elemento A	HEA120	HEA100
Perfil Elemento A	HEA120	HEA100
Ratio Elemento 0	0.7548	0.7564
Ratio Elemento 1	0.1439	0.1568
Ratio Elemento 2	0.1338	0.0509
Ratio Elemento 3	0.5186	0.1660
Ratio Elemento 4	0.5186	0.5291
Ratio Elemento 5	0.1338	0.5291
Ratio Elemento 6	0.1439	0.1660
Ratio Elemento 7	0.7548	0.0509
Ratio Elemento 8	0.1819	0.1568
Ratio Elemento 9	0.4110	0.7564
Ratio Elemento 10	0.5598	0.1109
Ratio Elemento 11	0.9932	0.2654
Ratio Elemento 12	0.9932	0.3913
Ratio Elemento 13	0.5598	0.4786
Ratio Elemento 14	0.4110	0.9101
Ratio Elemento 15	0.1819	0.9101
Ratio Elemento 16	0.4839	0.4786
Ratio Elemento 17	0.7238	0.3913
Ratio Elemento 18	0.6423	0.2654
Ratio Elemento 19	0.6852	0.1109
Ratio Elemento 20	0.7363	0.5014
Ratio Elemento 21	0.6852	0.8316
Ratio Elemento 22	0.6423	0.7759
Ratio Elemento 23	0.7238	0.7810
Ratio Elemento 24	0.4839	0.7723
Ratio Elemento 25	N/A	0.9172
Ratio Elemento 26	N/A	0.7723
Ratio Elemento 27	N/A	0.7810
Ratio Elemento 28	N/A	0.7759
Ratio Elemento 29	0.5834	0.8316
Ratio Elemento 30	0.6873	0.5014
Ratio Elemento 31	0.6620	N/A
Ratio Elemento 32	0.6276	N/A
Ratio Elemento 33	0.6245	N/A
Ratio Elemento 34	0.6620	N/A
Ratio Elemento 35	0.6873	N/A
Ratio Elemento 36	0.5834	0.5604
Ratio Elemento 37	N/A	0.752
Ratio Elemento 38	N/A	0.7366
Ratio Elemento 39	N/A	0.7273
Ratio Elemento 40	N/A	0.6184
Ratio Elemento 41	N/A	0.6184
Ratio Elemento 42	N/A	0.7273
Ratio Elemento 43	N/A	0.7366
Ratio Elemento 44	N/A	0.752
Ratio Elemento 45	N/A	0.5604

4.3 Tercer caso de estudio

Para el tercer caso de estudio se diseñó una viga armada simplemente apoyada considerando los siguientes parámetros:

- Luz por cubrir: 30 metros.
- Tipo de acero: S275.
- Carga distribuida de 80 kN/m.

Los elementos para la programación gráfica que se utilizaron para este caso se encuentran en la Figura 102. Comparado con el primer y segundo caso de estudio la programación gráfica para este caso es mucho menos densa en cantidad de complementos utilizados, pero más extensa en cuanto en cuanto al código utilizado, ya que este caso en su mayoría se realizó con programación en Python y Visual Basic. La optimización se realizó con Galápagos y Octopus de manera independiente.

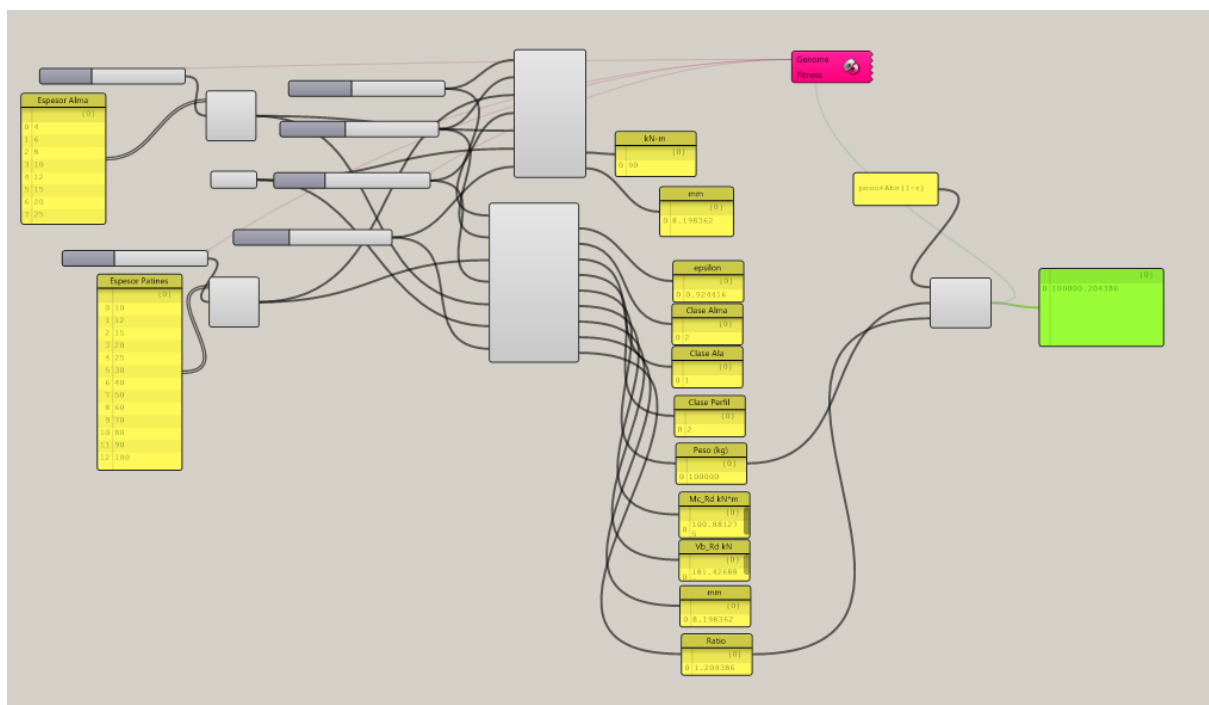


Figura 102. Programación Gráfica en Tercer Caso de Estudio

4.3.1 Criterio para mayoración de cargas

Las cargas se mayoraron de acuerdo con la Figura 44. Para estado límite último las cargas se mayoraron por 1.35 mientras que para estado límite de servicio las cargas se mayoraron utilizando el coeficiente de 1.0.

4.3.2 Modelación paramétrica

La Figura 103 muestra la programación gráfica definida en Grasshopper para obtener la geometría de la viga armada.

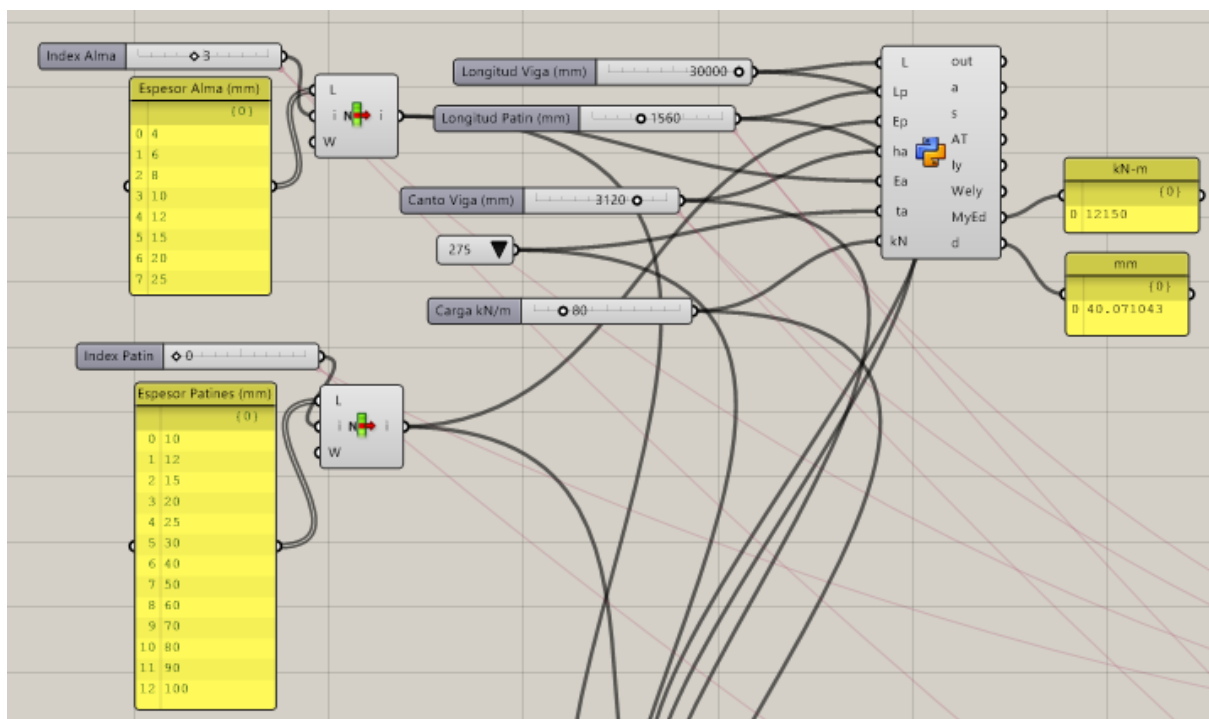


Figura 103. Modelación Geométrica de Viga Armada

A diferencia de los casos anteriores no hay componentes que definan la geometría si no que se colocó un solo componente llamado “geometría viga armada” (Ver Anejo 5) que a partir de parámetros generó un sólido que se muestra en la Figura 104 , Figura 105 y Figura 106.



Figura 104. Vista Frontal Viga Armada

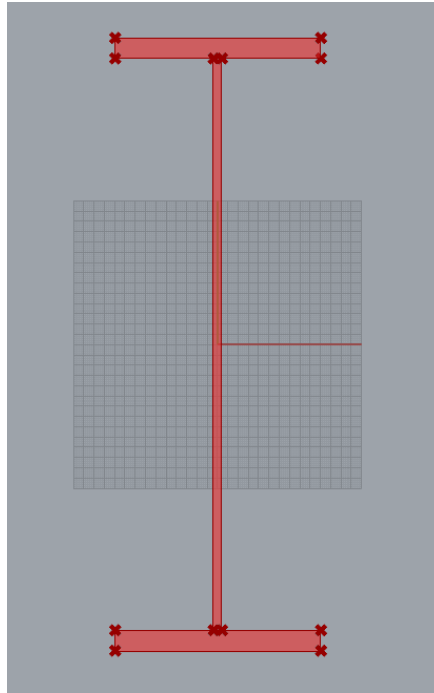


Figura 105. Sección Transversal Viga Armada

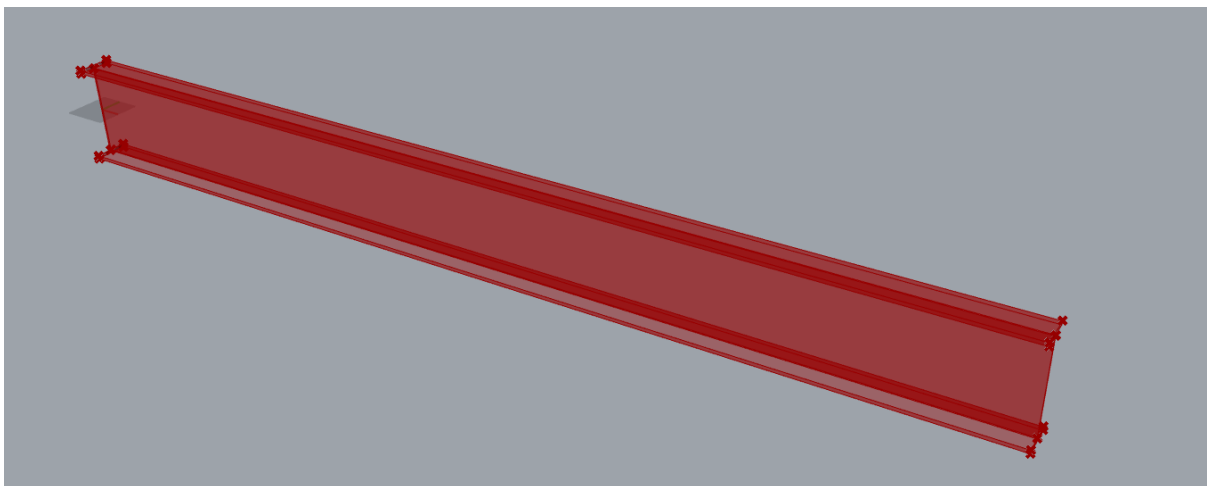


Figura 106. Vista en Perspectiva Viga Armada

El componente mostrado en la Figura 103 se nombró “geometría viga armada”, este componente se programó con lenguaje Python, dentro de la programación se trabaja con la librería de Rhinoceros 3D llamada Rhino.Geometry , está librería permite tener acceso a todos los comandos de Rhino que ayudan a modelar objetos tomando los elementos ya existentes.

La forma en que el componente “geometría viga armada” funciona es primero tomar los parámetros necesarios para generar la geometría, utiliza la longitud de la viga, ancho de la viga, espesor de los patines, canto de la viga, y espesor de la viga. Con los datos obtenidos primero genera los puntos que conforman el alma de la viga en el punto inicial de la viga sobre el eje x donde se tomó 0 como valor inicial. Con los cuatro puntos trazados del alma se generó la superficie del alma. Tras haber generado el alma se generaron los puntos del patín inferior y posteriormente su superficie. Al finalizar con el patín inferior se trazaron los puntos del patín superior y a partir de sus puntos se generó la superficie. Con la primera cara de la viga trazada se realizó el mismo procedimiento, pero tomando en cuenta el parámetro de longitud de la viga, por lo que la nueva cara se trazó a una distancia de 30 metros. Teniendo las 2 caras de la viga se generaron las superficies necesarias entre las 2 para genera finalmente un sólido.

4.4 Simulación estructural

El proceso de análisis se realizó utilizando un componente programado por código en Visual Basic llamado “análisis” de la Figura 107 (Ver Anejo 6). Los parámetros de entrada fueron la longitud de la viga, longitud del patín, espesor del patín, canto de la viga, espesor de la viga, tipo de acero y la carga aplicada.

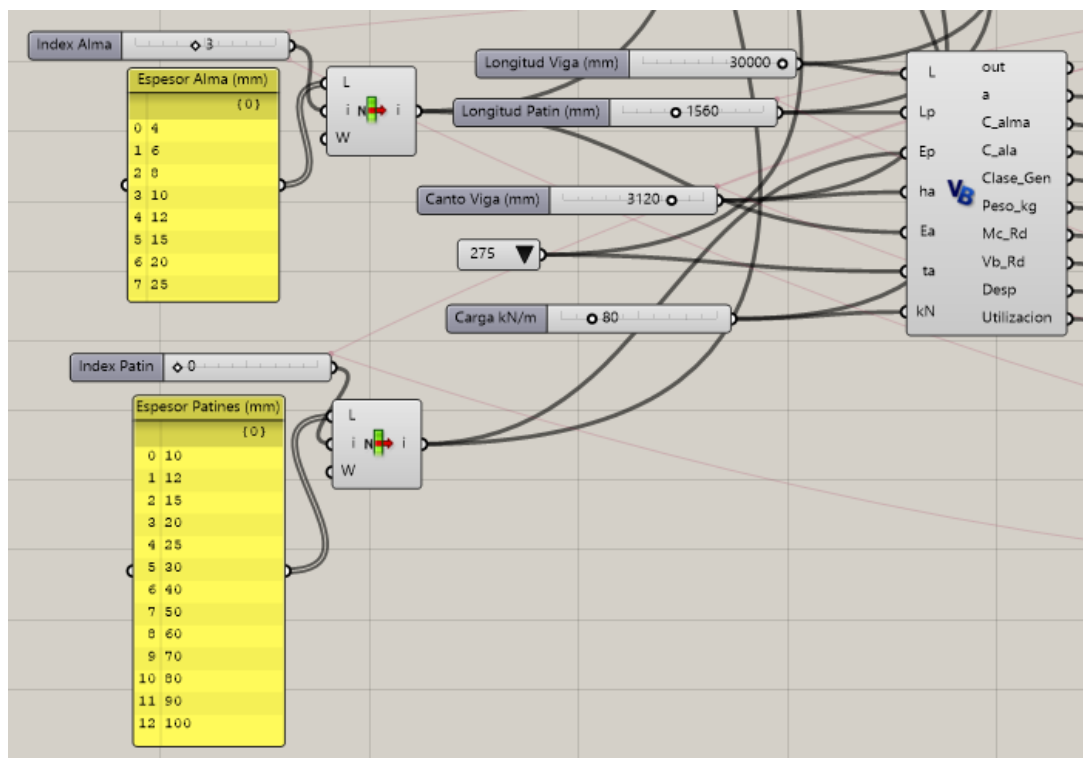


Figura 107. Componente en Código para Analizar Viga Armada

Al ser una viga con un perfil hecho a medida primero se comenzó por determinar la clase de la viga, obteniendo primero la clase del alma y de los patines obtenida en base a la tabla 5.2 mostrada en la Figura 108 y la tabla 5.3 de la figura Figura 109 respectivamente del Eurocódigo 3 parte 1-1 (EN 1993-1-1, 2005).

Internal compression parts						
				Axis of bending		
				Axis of bending		
Class	Part subject to bending	Part subject to compression	Part subject to bending and compression			
Stress distribution in parts (compression positive)						
1	$c/t \leq 72\varepsilon$	$c/t \leq 33\varepsilon$	when $\alpha > 0,5$: $c/t \leq \frac{396\varepsilon}{13\alpha - 1}$ when $\alpha \leq 0,5$: $c/t \leq \frac{36\varepsilon}{\alpha}$			
2	$c/t \leq 83\varepsilon$	$c/t \leq 38\varepsilon$	when $\alpha > 0,5$: $c/t \leq \frac{456\varepsilon}{13\alpha - 1}$ when $\alpha \leq 0,5$: $c/t \leq \frac{41,5\varepsilon}{\alpha}$			
Stress distribution in parts (compression positive)						
3	$c/t \leq 124\varepsilon$	$c/t \leq 42\varepsilon$	when $\psi > -1$: $c/t \leq \frac{42\varepsilon}{0,67 + 0,33\psi}$ when $\psi \leq -1^{*)}$: $c/t \leq 62\varepsilon(1 - \psi)\sqrt{(-\psi)}$			
$\varepsilon = \sqrt{235/f_y}$	f_y	235	275	355	420	460
	ε	1,00	0,92	0,81	0,75	0,71

*) $\psi \leq -1$ applies where either the compression stress $\sigma \leq f_y$ or the tensile strain $\varepsilon_y > f_y/E$

Figura 108. Relaciones Máximas Ancho/Espesor para Miembros a Compresión (Determinar Clase del Alma). Citado de (EN 1993-1-1, 2005)

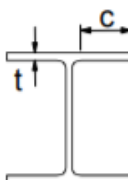
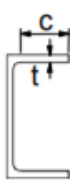
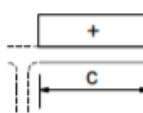
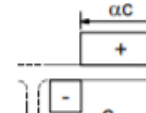
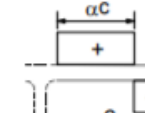
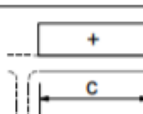
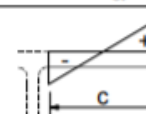
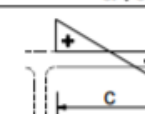
Outstand flanges						
						
Rolled sections			Welded sections			
Class	Part subject to compression	Part subject to bending and compression				
		Tip in compression		Tip in tension		
Stress distribution in parts (compression positive)						
1	$c/t \leq 9\epsilon$	$c/t \leq \frac{9\epsilon}{\alpha}$		$c/t \leq \frac{9\epsilon}{\alpha\sqrt{\alpha}}$		
2	$c/t \leq 10\epsilon$	$c/t \leq \frac{10\epsilon}{\alpha}$		$c/t \leq \frac{10\epsilon}{\alpha\sqrt{\alpha}}$		
Stress distribution in parts (compression positive)						
3	$c/t \leq 14\epsilon$	$c/t \leq 21\epsilon\sqrt{k_{\sigma}}$				
For k_{σ} see EN 1993-1-5						
$\epsilon = \sqrt{235/f_y}$	f_y	235	275	355	420	460
	ϵ	1,00	0,92	0,81	0,75	0,71

Figura 109. Relaciones Máximas Ancho/Espesor para Miembros a Compresión (Determinar Clase de Patín). Citado de (EN 1993-1-1, 2005)

Tras seleccionar la clase del perfil en caso de ser clase 4 se procedió a obtener el factor de reducción de área por pandeo en placas llamado ρ , este procedimiento se realizó de acuerdo con las ecuaciones descritas a continuación tomadas del Eurocódigo 3 parte 1-5 (EN 1993-1-5, 2006):

- Área efectiva:

$$A_{c,eff} = \rho A_c \quad \text{Ecuación (4.1)}$$

- Cuando $\lambda p \leq 0.673$ el valor de ρ se toma como 1, cuando $\lambda p > 0.673$ se realiza la siguiente ecuación:

$$\rho = \frac{\lambda p - 0.055(3 + \psi)}{\lambda p^{-2}} \leq 1 \quad \text{Ecuación (4.2)}$$

- El valor de λp se obtiene a partir de la siguiente ecuación:

$$\lambda p = \frac{b/t}{28.4 \cdot \varepsilon \cdot \sqrt{k_\sigma}}; \varepsilon = \sqrt{\frac{235}{f_y}}$$

- El valor de ψ y k_σ se obtiene de la tabla 4.1 del Eurocódigo 3 parte 1-5 (EN 1993-1-5, 2006) mostrado en la Figura 110. Estos valores dependen de la distribución de esfuerzos provocada por la carga aplicada y condiciones de contorno, al ser la carga distribuida y una viga simplemente apoyada el valor de ψ se tomó como -1 y el valor de k_σ se tomó como 23.9.

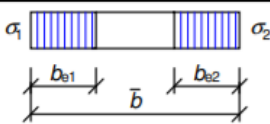
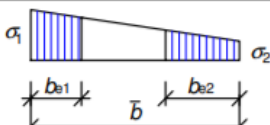
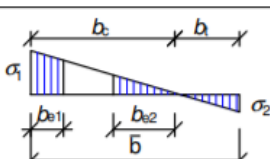
Stress distribution (compression positive)				Effective ^p width b_{eff}		
				$\psi = 1:$ $b_{eff} = \rho \bar{b}$ $b_{e1} = 0,5 b_{eff} \quad b_{e2} = 0,5 b_{eff}$		
				$1 > \psi \geq 0:$ $b_{eff} = \rho \bar{b}$ $b_{e1} = \frac{2}{5 - \psi} b_{eff} \quad b_{e2} = b_{eff} - b_{e1}$		
				$\psi < 0:$ $b_{eff} = \rho b_c = \rho \bar{b} / (1 - \psi)$ $b_{e1} = 0,4 b_{eff} \quad b_{e2} = 0,6 b_{eff}$		
$\psi = \sigma_2 / \sigma_1$	1	$1 > \psi > 0$	0	$0 > \psi > -1$	-1	$-1 > \psi > -3$
Buckling factor k_σ	4,0	$8,2 / (1,05 + \psi)$	7,81	$7,81 - 6,29\psi + 9,78\psi^2$	23,9	$5,98 (1 - \psi)^2$

Figura 110. Relación entre Factores de Pandeo y Esfuerzo en Base a Condiciones de Contorno. Citado de (EN 1993-1-5, 2006)

- La resistencia a cortante se consideró como la contribución del cortante resistente por el alma más la contribución del cortante resistente por los patines.

La contribución de resistencia a cortante por el alma se calculó como:

$$V_{bw,Rd} = \frac{\chi_w f_{yw} h_w t}{\sqrt{3} Y_{M1}} \quad (\text{Ecuación 5.2})$$

El valor de χ_w se tomó de la tabla 5.1 del Eurocódigo 3 parte 1-5 en la figura Figura 111 (EN 1993-1-5, 2006) y el valor de Y_{M1} se tomó como 1.

	Rigid end post	Non-rigid end post
$\bar{\lambda}_w < 0,83/\eta$	η	η
$0,83/\eta \leq \bar{\lambda}_w < 1,08$	$0,83/\bar{\lambda}_w$	$0,83/\bar{\lambda}_w$
$\bar{\lambda}_w \geq 1,08$	$1,37/(0,7 + \bar{\lambda}_w)$	$0,83/\bar{\lambda}_w$

Figura 111. Contribución del alma χ_w a Resistencia a Pandeo por Cortante

La contribución de resistencia a cortante por los patines se consideró como:

$$\tau_{cr} = k_\tau \sigma_E \quad (\text{Ecuación 5.4})$$

Donde

$$\sigma_E = \frac{\pi^2 E t^2}{12(1 - \nu^2) b^2} = 190000 \left(\frac{t}{b} \right)^2 \quad (\text{Ecuación A.1})$$

Cuando $a/h_w \geq 1$

$$k_\tau = 5.34 + 4.00(h_w/a)^2 + k_{\tau sl} \quad (\text{Ecuación A.5})$$

Cuando $a/h_w < 1$

$$k_\tau = 4.00 + 5.34(h_w/a)^2 + k_{\tau sl} \quad (\text{Ecuación A.5})$$

$$k_{\tau sl} = 9 \left(\frac{h_w}{a} \right)^2 \sqrt[4]{\left(\frac{I_{sl}}{t^3 h_w} \right)^3} \text{ pero no menos que } \frac{2.1}{t} \sqrt[3]{\frac{I_{sl}}{h_w}} \quad (\text{Ecuación A.5})$$

Para la resistencia a compresión, momento y pandeo lateral se tomaron las ecuaciones del Eurócodigo 3 parte 1-1 (EN 1993-1-1, 2005):

- La resistencia a compresión de la viga para clase 1, 2 y 3 se obtuvo con:

$$N_{c,Rd} = \frac{A f_y}{\gamma_{M0}} \quad (\text{Ecuación 6.10})$$

- Para perfiles clase 4 la resistencia a compresión se obtuvo a partir de:

$$N_{c,Rd} = \frac{A_{eff} f_y}{\gamma_{M0}} \quad (\text{Ecuación 6.11})$$

- La resistencia por flexión para clase 1 y 2 se obtuvo con:

$$M_{c,Rd} = \frac{W_{pl} f_y}{\gamma_{M0}} \quad (\text{Ecuación 6.13})$$

- La resistencia por flexión para clase 3 se obtuvo con:

$$M_{c,Rd} = \frac{W_{el} f_y}{\gamma_{M0}} \quad (\text{Ecuación 6.14})$$

- La resistencia por flexión para clase 4 se obtuvo con:

$$M_{c,Rd} = \frac{W_{eff,min} f_y}{\gamma_{M0}} \quad (\text{Ecuación 6.15})$$

- Para pandeo lateral se utilizó la ecuación:

$$M_{b,Rd} = \chi_{LT} W_y \frac{f_y}{\gamma_{M1}} \quad (\text{Ecuación 6.55})$$

-Para clase 1 y 2 $W_y = W_{pl,y}$

-Para clase 3 $W_y = W_{el,y}$

-Para clase 4 $W_y = W_{eff,y}$

-El valor de reducción para pandeo lateral χ_{LT} se obtuvo con:

$$\chi_{LT} = \frac{1}{\Phi_{LT} + \sqrt{\Phi_{LT}^2 - \bar{\lambda}_{LT}^2}}, \chi_{LT} \leq 1.0 \quad (\text{Ecuación 6.56})$$

$$\Phi_{LT} = 0.5[1 + \alpha_{LT}(\bar{\lambda}_{LT} - 0.2) + \bar{\lambda}_{LT}^2] \quad (\text{Ecuación 6.56})$$

-El valor para el factor de imperfección α_{LT} se obtuvo a partir de la tabla 6.3 y 6.4 del Eurocódigo 3 parte 1-1 (EN 1993-1-1, 2005) mostrado en la Figura 112 y Figura 113.

Buckling curve	a	b	c	d
Imperfection factor α_{LT}	0,21	0,34	0,49	0,76

Figura 112. Valores Recomendados para Factores de Imperfección para Curvas de Pandeo Lateral.

Citado de (EN 1993-1-1, 2005)

Cross-section	Limits	Buckling curve
Rolled I-sections	$h/b \leq 2$	a
	$h/b > 2$	b
Welded I-sections	$h/b \leq 2$	c
	$h/b > 2$	d
Other cross-sections	-	d

Figura 113. Valores Recomendados para Curvas de Pandeo Lateral para Secciones Transversales. Citado

de (EN 1993-1-1, 2005)

-El valor de $\bar{\lambda}_{LT}$ se obtuvo a partir de:

$$\bar{\lambda}_{LT} = \sqrt{\frac{W_y f_y}{M_{cr}}} \quad (\text{Ecuación 6.56})$$

-El valor de M_{cr} se obtuvo con:

$$M_{cr} = C_1 \frac{\pi^2 E I_z}{L^2} \sqrt{\frac{I_w}{I_z} + \frac{L^2 G I_t}{\pi^2 E I_z}}$$

$$C_1 = \frac{1}{(k_c)^2}$$

-El valor de corrección k_c se obtuvo a partir de la tabla 6.6 del Eurocódigo 3 parte 1-1 (EN 1993-1-1, 2005) en la Figura 114. Se tomó el valor de 0.94 debido a la distribución de momentos por carga distribuida y por ser una viga simplemente apoyada. La Figura 111 se podría colocar como parámetro de entrada para considerar los distintos tipos de distribución de momentos y dar mayor versatilidad al sistema.


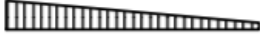






Moment distribution	k_c
 $\psi = 1$	1,0
 $-1 \leq \psi \leq 1$	$\frac{1}{1,33 - 0,33\psi}$
	0,94
	0,90
	0,91
	0,86
	0,77
	0,82

Figura 114. Factores de Corrección K_c . Citado de (EN 1993-1-1, 2005)

El momento, cortante y el desplazamiento se obtuvieron tomando como base una viga simplemente apoyada con carga distribuida de acuerdo con las siguientes ecuaciones:

$$M_{max} = \frac{qL^2}{8}$$

$$V_{max} = \frac{qL}{2}$$

$$Flecha_{max} = \frac{5qL^4}{384EI}$$

De acuerdo al Eurocódigo 3 parte 1-1 (EN 1993-1-1, 2005) verificaron los ratios de utilización con las siguientes ecuaciones:

$$\frac{M_{Ed}}{M_{c,Rd}} \leq 1.0 \quad (Ecuación 6.12)$$

$$\frac{V_{Ed}}{V_{c,Rd}} \leq 1.0 \quad (Ecuación 6.1)$$

$$\frac{M_{Ed}}{M_{b,Rd}} \leq 1.0 \quad (Ecuación 6.54)$$

Al ser una viga con carga distribuida y simplemente apoyada no se consideraron combinaciones entre flexión y cortante ya que en el punto de máximo momento el cortante es cero y en el punto de máximo cortante el momento es cero. Para que el perfil verificara debió cumplir con el máximo de los ratios de utilización obtenidos.

4.4.1 Optimización estructural

El proceso de optimización se realizó con Galágos y con Octopus de manera independiente para realizar una comparación entre ambos componentes. El elemento óptimo se consideró como el menos pesado, que cumpla con la flecha establecida y más cercano al 1 en cuanto a su ratio de utilización.

4.4.2 Optimización con Galápagos

Los parámetros o “Genome” que se modificaron para realizar la optimización se encuentran en la Figura 115, estos parámetros incluyen la longitud y espesor del patín, canto de la viga y espesor del alma. Al ser los espesores limitados por fabricación se colocaron en una lista los espesores más comunes para que la selección se hiciera a partir de ellos.

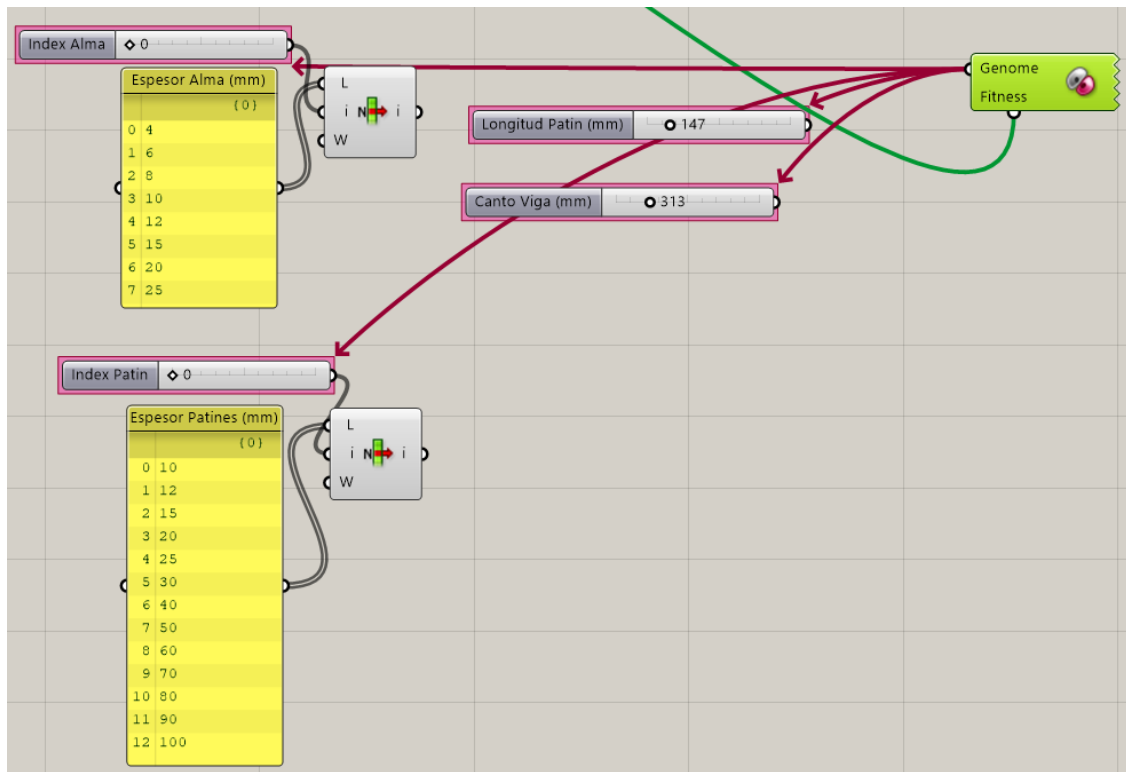


Figura 115. Parámetros para Modificar por Galápagos en Optimización

Como objetivo se estableció una función que suma el peso de la estructura, el inverso del ratio de utilización y la suma del desplazamiento, se observa en la Figura 116.

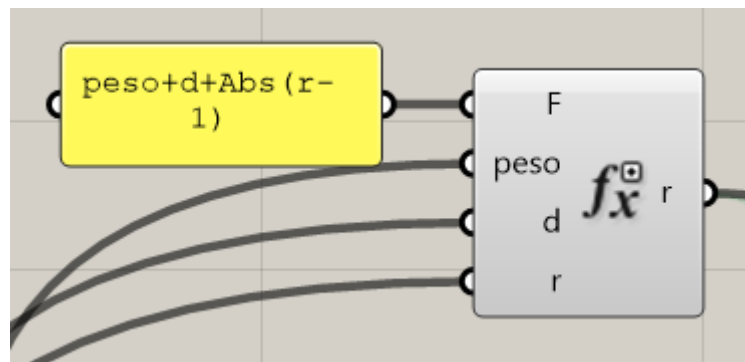


Figura 116. Función de Optimización Tercer Caso de Estudio

La verificación del desplazamiento se realiza de forma interna en el componente programado de la Figura 107. Lo que hace el componente es verificar si la flecha cumple con el parámetro de $L/300$, de ser así no hay modificaciones, en caso de no cumplir con la limitante de desplazamiento el código modifica desplazamiento de la estructura a 100.000 mm, este parámetro se encuentra relacionado con la función de la Figura 116, por lo que de manera directa se verifica la flecha durante el proceso de optimización. Otra verificación que se hizo

fue sobre el ratio de utilización, esta verificación también se programó en el componente de la Figura 107, en caso de tener un ratio mayor a 1 el ratio se modifica a 100.000 y en caso de ser menor o igual a 1 el ratio se mantiene en su estado original, de esta manera de nuevo se mantiene una verificación constante durante el proceso de optimización por Galápagos.

Un aspecto relevante durante la variación de los parámetros de la viga fue mantener una proporción adecuada entre el ancho de los patines y el canto de la viga, en el componente de la Figura 107 se realizó una revisión en el cual se debe mantener una proporción de $1 \leq h_w/bf \leq 2$ en caso de no cumplir el peso se modifica nuevamente a 100.000 kg y en caso de cumplir el peso se mantiene en su estado original, de esta forma durante el proceso iterativo de Galápagos se mantiene una proporción adecuada en la viga armada.

Los resultados obtenidos se encuentran en la Figura 117 y resumidos en la Tabla 16.

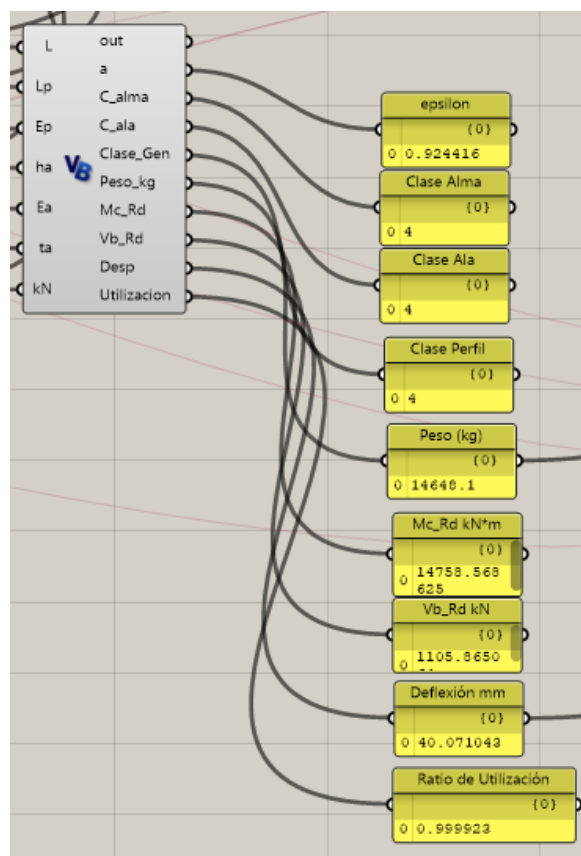


Figura 117. Resultados Optimización Galápagos en Tercer Caso de Estudio

Tabla 16. Resumen de Resultado Optimización Galápagos Tercer Caso de Estudio

Resultado Optimización Galápagos	
Canto (mm)	3120
Ancho patines (mm)	1560
Espesor alma (mm)	10
Espesor patines (mm)	10
Clase del Perfil	4
Peso (kg)	14648
Flecha (mm)	40
Ratio de Utilización	0.999
Número de Iteraciones	26000
Tiempo de Optimización (s)	266

4.4.3 Optimización con Octopus

La optimización con Octopus se realizó utilizando los mismos parámetros de la Figura 115 y la misma función de optimización de la Figura 116, simplemente se intercambié el componente de Galápagos por el de Octopus para analizar los resultados obtenidos a través de este optimizador. El resultado obtenido se muestra en la Figura 118 y el resumen de resultados en la Tabla 17.

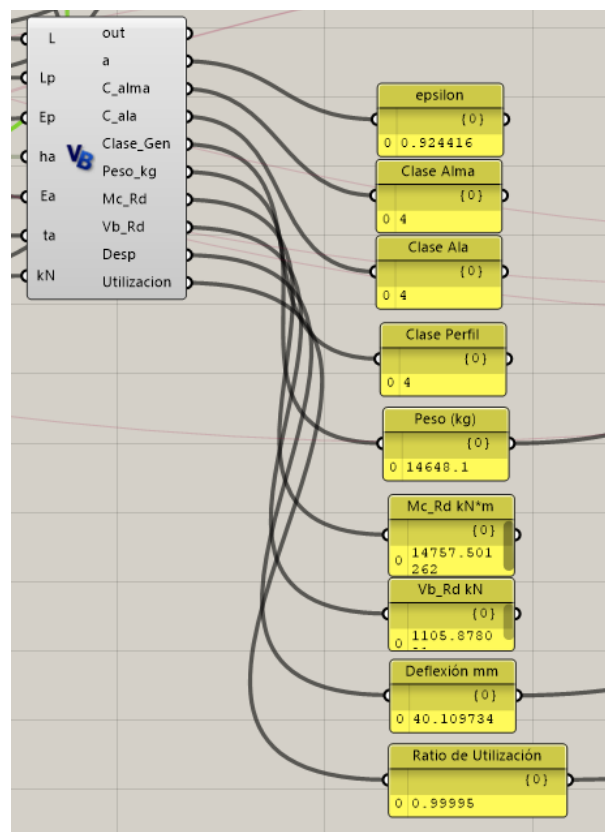


Figura 118. Resultados Optimización Octopus en Tercer Caso de Estudio

Tabla 17. Resumen de Resultado Optimización Octopus Tercer Caso de Estudio

Resultado Optimización Octopus	
Canto (mm)	3118
Ancho patines (mm)	1561
Espesor alma (mm)	10
Espesor patines (mm)	10
Clase del Perfil	4
Peso (kg)	14648
Flecha (mm)	40
Ratio de Utilización	0.999
Número de Iteraciones	25650
Tiempo de Optimización (s)	900

Para comparar ambos resultados se generó la Tabla 18.

Tabla 18. Comparación de Resultados entre Galápagos y Octopus

Resultados de proceso de optimización			
	Galápagos	Octopus	Diferencia %
Canto (mm)	3120	3118	-0.06
Ancho patines (mm)	1560	1561	0.06
Espesor alma (mm)	10	10	0.00
Espesor patines (mm)	10	10	0.00
Clase del Perfil	4	4	0.00
Peso (kg)	14648	14648	0.00
Flecha (mm)	40	40	0.00
Ratio de Utilización	0.999	0.999	0.00
Número de Iteraciones	26000	25650	-1.36
Tiempo de Optimización (s)	266	900	70.44

5 Discusión de resultados

El primer caso de estudio de la viga simplemente apoyada con una carga puntual al centro y una carga axial demuestra que el optimizar de perfiles con el que cuenta Karamba 3D funciona de manera adecuada, dependiendo el catálogo de perfiles que se ponga a disposición del componente el selector toma el perfil menos pesado y con un ratio de utilización mayor. El componente de optimización cuenta con ciertas ventajas como el realizar las verificaciones de manera automática de acuerdo al Eurocódigo 3, verifica las distintas interacciones entre flexión y cortante, flexión y axil y hace la verificación del perfil obteniendo los factores de interacción en base al método 2 del Eurocódigo, hay ocasiones en que en proyectos reales dependiendo del cliente y el proyecto surge la necesidad de obtener los factores con un método distinto y el no tener la disponibilidad de seleccionarlo es una desventaja. Otra de las desventajas que se observaron en general es que los casos de carga se deben introducir de manera manual y que a pesar de tener la posibilidad de introducir los casos de carga, al revisar la flecha del elemento se verifica por la flecha más grande presente en los casos de carga y no por la flecha generada por estado límite de servicio. El complemento de optimización de perfiles obtiene buenos resultados para casos en que se debe modificar los elementos de forma individual, pero al buscar una optimización más compleja o de elementos que trabajan en conjunto no sirve para el propósito.

El objetivo del segundo caso es optimizar una celosía con una carga puntual al centro y con los apoyos empotrados. Para la optimización se utilizó Galápagos y una combinación entre Galápagos y Octopus. Comparado al primer caso de estudio la optimización que se realizó fue más compleja ya que el objetivo no solo fue optimizar los perfiles, sino que también la geometría de la estructura, de esta manera se demostró que es posible trabajar casos de optimización más complejos gracias a los complementos Galápagos y Octopus, pero se requiere establecer componentes personalizados con programación para limitar las soluciones y generar resultados más óptimos. Al estar trabajando con los datos de forma más libre fue posible decidir el caso de carga deseado para verificar flechas y el caso de carga deseado para verificar resistencia y estabilidad por lo que las limitaciones del primer caso de estudio desaparecieron. Una de las limitantes de utilizar Galápagos es que solo verifica un objetivo, se puede utilizar para verificar muchos objetivos como se hizo, pero la dificultad se encuentra en encontrar una manera de unir los objetivos en uno solo. La ventaja de Octopus es que es multiobjetivo por lo

que no es difícil seleccionar los objetivos a utilizar, la desventaja en este caso de estudio es que uno de los objetivos era los ratios de utilización en forma de lista y Octopus no los podía leer de esta manera, es por eso que se tuvo que combinar con Galápagos. En cuanto a los resultados obtenidos con el componente Galápagos se obtuvo una celosía con más elementos que con la combinación de Octopus y Galapagos pero el peso obtenido fue menor, logrando un mejor objetivo. Una de las razones por lo que el resultado de Octopus fue peor es porque no se pudo manejar de manera individual y la celosía se optimizó en conjunto con Galápagos. Es posible que de haber encontrado una manera de realizar la optimización con Octopus de manera individual el resultado podría mejorar.

El tercer caso de estudio trató de una viga simplemente apoyada con carga distribuida, en este caso se buscó optimizar una viga armada, comparado a los casos anteriores es un caso más complejo ya que el perfil a revisar no es un existente y no se pudieron utilizar los componentes de Karamba 3D. Entre más complejo es el caso más se requieren componentes personalizados con un mayor nivel de programación para obtener los resultados deseados. En cuanto a la optimización con Galápagos y Octopus se podría decir que llegaron prácticamente al mismo resultado, el parámetro que más variación tuvo fue el tiempo de optimización ya que tardó un 70% más que Galápagos en obtener el resultado, no quiere decir que Galápagos sea mejor, pero en este caso de estudio en particular los resultados de Galápagos fueron más favorables.

Fue muy importante la interacción BIM que hubo dentro de los programas ya que esto permitió una interconectividad e interoperabilidad muy fluida entre realizar las geometrías de los casos de estudio y visualizar los modelos al igual que permitió operabilidad entre la geometría generada y el análisis estructural, al crear elementos de este tipo existe la posibilidad de exportar el modelo generado y analizarlo en programas de cálculo externos que admitan el mismo tipo de archivos y no existe la limitación de trabajar forzosamente con los complementos de Karamba 3D o de programar para analizar los elementos.

6 Conclusiones y recomendaciones

Los tres casos de estudio de esta tesis se desarrollaron con el objetivo de desarrollar una metodología que aprovechara tanto el diseño paramétrico como los algoritmos genéticos para optimizar una estructura, aunque los casos de estudio son sencillos esta metodología se puede aplicar a estructuras más complejas, la variante se encuentra en el modelo paramétrico que se va a utilizar.

El primer caso de estudio el cual utiliza el componente de Karamba 3D para su optimización demuestra que para casos aislados se puede optimizar una estructura pero no se tiene la flexibilidad para que el usuario decida exactamente sobre que parámetros optimizar el elemento, es decir si quiere priorizar la flecha sobre el peso no podrá, mientras que en el caso de estudio dos y tres se utilizan algoritmos genéticos, tanto Galápagos como Octopus mostraron dar al usuario una mayor flexibilidad sobre los parámetros objetivo en los cuales basar la optimización.

Es una gran ventaja contar con la interoperabilidad que brindan los componentes a trabajar en un entorno como Grasshopper ya que hay a la mano muchas herramientas con las cuales trabajar y esto brinda una gran flexibilidad al usuario sobre los proyectos, un ejemplo de esto es que en el segundo caso de estudio la simulación estructural se realizó con Karamba 3D, mientras que en el tercer caso de estudio no se usó Karamba 3D pero al tener distintas opciones y flexibilidad dentro de Grasshopper se pudo utilizar programación en código para sustituir esta necesidad.

Durante la etapa de aprendizaje de Karamba 3D se encontró que en algunos modelos con cierta distribución de cargas se generaba una reducción de la capacidad del elemento a compresión y a cortante al realizar sobre si el cortante actuante excede el 50% del cortante resistente, se daban casos en donde por este proceso los extremos de un elemento tenían una resistencia menor a las secciones internas, en los elementos analizados en esta tesis no ocurren esos problemas. Se mantuvo contacto con el desarrollador y comentó que estaba al tanto, comentó que no había mucho problema ya que el resultado se encontraba por el lado de la seguridad pero que en la próxima actualización se iba a solucionar.

En cuanto al proceso de actualización se recomienda aplicar el procedimiento de Galápagos a una estructura con múltiples elementos para probar la optimización con un grupo de elementos de forma simultánea. También se recomienda encontrar una manera de que Octopus pueda trabajar con listas como objetivo para que se pueda probar de manera independiente y obtener mejores resultados con este complemento.

En cuanto al tercer caso de estudio se puede realizar un próximo trabajo en donde se amplié la programación y se pueda analizar utilizando el método de elementos finitos, de igual manera sería recomendable probar la optimización con más vigas armadas trabajando en conjunto.

En algún proyecto futuro se puede estudiar que tan bien se comportan los elementos geométricos generados en Grasshopper en programas de análisis estructural externos. El hecho de que todos los componentes de programación visual se encuentren fácilmente vinculados a software de gestión BIM hace especialmente atractivo su uso en los próximos años en los que se entenderá el BIM como un estándar en el sector de ingeniería y construcción.

7 Bibliografía

- Ansys. (1970). Obtenido de <https://www.ansys.com/>
- Autodesk. (s.f.). *Revit*. Obtenido de <https://www.autodesk.es/>
- Bendsøe, M. (1989). *Optimal shape design as a material distribution problem*. *Struct Multidiscip Optim.* 1, 193-202.
- Cerrolaza, M., & Annicchiarico, W. (1996). *Algoritmos de optimización estructural basados en simulación genética*. Caracas: Universidad Central de Venezuela. Primera Edición.
- Costanzi, C., Ahmed, Z., Schipper, H., Bos, F., Knaack, U., & Wolfs, R. (2018). 3D Printing Concrete on temporary surfaces: The design and fabrication of a concrete shell structure. *Automation in Construction*, 94, 395-404.
- Davis, D. (2013). *Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture*. PhD dissertation. RMIT University.
- EN 1990. (2002). *Eurocode: Basis of structural design*. Bruselas: Comisión Europea de normalización CEN.
- EN 1993-1-1. (2005). *Eurocode 3: Design of Steel Structures. Part 1-1: General rules and rules for buildings*. Bruselas: Comisión Europea de normalización CEN.
- EN 1993-1-5. (2006). *Eurocode 3: Design of Steel Structures. Part 1-5: Plated structural elements*. Bruselas: Comisión Europea de normalización CEN.
- EnergyPlus. (s.f.). Obtenido de <https://energyplus.net/>
- Gil, A., & Parsons, R. (2014). *The Grasshopper Primer Third Edition*. Mode Lab.
- Gulvanessian, H., & Holicky, M. (1996). *Designer's Handbook to Eurocode 1 Part 1: Basis of design*. London: Thomas Telford. 1st Edition.
- Ibrahim, H., Wagdy, A., Beccarelli, P., Carpenter, R., & Chilton, J. (2016). Applicability of Flexible Photovoltaic Modules onto Membrane Structures Using Grasshopper Integrative Model. *Procedia Engineering*, 155, 379-387.
- Kensek, K. (2014). Integration of Environmental Sensors with BIM: case studies. *Informes de la Construcción*, 66, 536-544.
- Kravanja, S., Turkalj, G., Silih, S., & Zula, T. (2013). Optimal design of single-story steel building structures based on parametric MINLP optimization. *Journal of Constructional Steel Research*, 81, 86-103.
- Lee, S., & Ha, M. (2013). Customer interactive building information modeling for apartment unit design. *Automation in Construction*, 35, 424-430.
- McNeel & Associates. (2019). *Rhinoceros*. Obtenido de <https://www.rhino3d.com/6/features>

- McNeel, R. (2019). *McNeel Wiki*. Obtenido de <https://wiki.mcneel.com/rhino/nurbssurfaces>
- Muttio, E., Botello, S., & Tapia, M. (2017). Modelado paramétrico mediante programación visual en el diseño y análisis estructural de edificios. *Revista Mexicana de Métodos Numéricos, 1*. Obtenido de https://www.scipedia.com/public/Muttio_et_al_2017a
- Pan, W., Turrin, M., Louter, C., Sariyildiz, S., & Sun, Y. (2019). Integrating multi-functional space and long-span structure in the early design stage of indoor sports arenas by using parametric modelling and multi-objective optimization. *Journal of Building Engineering, 22*, 464-485.
- Paulson, B. (1976). Designing to Reduce Construction Costs. *Journal of the Construction Division, 102*, 587–592.
- Preisinger, C. (2013). *Linking Structure and Parametric Geometry*. Obtenido de *Archit Design, 83*: 110-113.
- Preisinger, C. (2019). *Karamba 3D Parametric Engineering*. Obtenido de User Manual (Version 1.3.2): <https://www.karamba3d.com/download/>
- Reinhart, C. (1998). *Daysim*. Obtenido de <https://daysim.ning.com/>
- Rodríguez, A. (2015). *Grasshopper - Pabellón Paramétrico [LunchBox]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=MqDIODqGkZ8>
- Rombouts, J., Lombaert, G., De Laet, L., & Schevenels, M. (2019). A novel shape optimization approach for strained gridshells: Design and construction of a simply supported gridshell. *Engineering Structures, 192*, 166-180.
- Ruiz, R. (2015). *Arquitectura de voltereta*. Obtenido de WordPress: <https://arquitecturadevoltereta.wordpress.com/>
- Rutten, D. (September de 2010). *Evolutionary Principles applied to Problem Solving using Galapagos*. Obtenido de <https://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>
- s.p.a, E. (2002). *modeFRONTIER*. Obtenido de <https://en.wikipedia.org/wiki/ModeFRONTIER>
- Sánchez, S. (2012). Optimización estructural y topológica de estructuras morfológicamente no definidas mediante algoritmos genéticos. Tesis Doctoral. *Universidad Politécnica de València*. Obtenido de doi:10.4995/Thesis/10251/15409
- Simonetti, H. L., Almeida, V. S., & Neto, L. d. (2014). A smooth evolutionary structural optimization procedure applied to plane stress problem. *Engineering Structures, 75*, 248-258.

- Simonetti, H. L., Almeida, V., Leitao, V., & Neves, F. (2019). Application of evolutionary structural optimization based on the boundary element method using initial stress fields. *Structures*, 20, 550-559.
- Sörensen, K., Sevaux, M., & Glover, F. (2018). *A History of Metaheuristics*. In: Martí R., Panos P., Resende M. (eds) *Handbook of Heuristics*. Springer.
- Taleb, H., & Musleh, M. (2015). Applying urban parametric design optimisation processes to a hot climate: Case study of the UAE. *Sustainable Cities and Society*, 14, 236-253.
- Tekla. (s.f.). Obtenido de <https://www.tekla.com/la>
- Tomei, V., Imbimbo, M., & Mele, E. (2018). Optimization of structural patterns for tall buildings: The case of diagrid. *Engineering Structures*, 171, 280-297.
- Vierlinger, R. (December de 2012). *Octopus*. Obtenido de Grasshopper: <https://www.grasshopper3d.com/group/octopus?groupUrl=octopus&id=2985220%3AGroup%3A742529&page=7#comments>
- Yang, D., Ren, S., Turrin, M., Sariyildiz, S., & Sun, Y. (2018). Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design. *Automation in Construction*, 92, 242-269.

Anexos

Anejo 1. Código en lenguaje de programación Visual Basic del componente “limitar canto”

```
'Limita el canto máximo
If Canto >= L / 15 And Canto <= L / 10 Then
    A = P
Else
    A = 10000
End If
```

Anejo 2. Código en lenguaje de programación Visual Basic del componente “limitar desplazamiento”

```
'Limita Desplazamiento a L/300
If Desp > (L / 300 * 100) Then
    A = 1000
Else
    A = Desp
End If
```

Anejo 3. Código en lenguaje de programación Visual Basic del componente “verifica ratio”

```
'Convierte a 1000 los valores mayores a 1
If x > 1 Then
    x = 1000
End If
```

Anejo 4. Código en lenguaje de programación Visual Basic del componente “convertir a mil”

```
'Si hay algún valor mayor a 1000 convierte a todos a 1000
Dim f As Double
f = 0
For i As Integer = 0 To (L - 1)
    If x(i) = 1000 Then
        f = 1
    End If
Next
If f = 1 Then
    For i As Integer = 0 To (L - 1)
        x(i) = 1000
    Next
End If
A = x
```

Anejo 5. Código en lenguaje de programación Python del componente “geometría viga armada”

```
import rhinoscriptsyntax as rs
import Rhino.Geometry as rg
import math

#Generar puntos alma en x=0

p1=rg.Point3d(0,-Ea/2,ha/2-Ep)
p2=rg.Point3d(0,Ea/2,ha/2-Ep)
p3=rg.Point3d(0,-Ea/2,-ha/2-Ep)
p4=rg.Point3d(0,Ea/2,-ha/2-Ep)
a=p1,p2,p3,p4

#Generar Superficie 1

s1=rg.NurbsSurface.CreateFromCorners(p1,p2,p4,p3)
#Cálculo de Área 1
A1=Ea * (ha-2*Ep)

#Generar puntos patín inf en x=0

p5=rg.Point3d(0,-Lp/2,-ha/2+Ep)
p6=rg.Point3d(0,-Lp/2,-ha/2)
p7=rg.Point3d(0,Lp/2,-ha/2)
p8=rg.Point3d(0,Lp/2,-ha/2+Ep)

#Generar Superficie 2

s2=rg.NurbsSurface.CreateFromCorners(p5,p6,p7,p8)
#Calculo de Área 2
A2=Lp *Ep

#Generar puntos patín sup en x=0

p9=rg.Point3d(0,-Lp/2,ha/2)
p10=rg.Point3d(0,-Lp/2,ha/2-Ep)
p11=rg.Point3d(0,Lp/2,ha/2-Ep)
p12=rg.Point3d(0,Lp/2,ha/2)

#Generar Superficie 2

s3=rg.NurbsSurface.CreateFromCorners(p9,p10,p11,p12)

#Cálculo de Área 3
A3=Lp *Ep
```

#Cálculo de Área Total

$AT=A1+A2+A3$

#Cálculo de Inercia en Y

$div1=ha/2-Ep$

$I1=(Ea*\text{pow}(ha-2*Ep,3)/12)+A1*\text{pow}(0,2)$

$I2=(Lp*\text{pow}(Ep,3)/12)+A2*\text{pow}(ha/2-Ep/2,2)*2$

$Iy=I1+I2$

#Calculo Modulo Resistente en Y

$Wely=Iy/(ha/2)$

#Generar puntos alma en x=L

$p13=rg.\text{Point3d}(L,-Ea/2,ha/2-Ep)$

$p14=rg.\text{Point3d}(L,Ea/2,ha/2-Ep)$

$p15=rg.\text{Point3d}(L,-Ea/2,-ha/2-Ep)$

$p16=rg.\text{Point3d}(L,Ea/2,-ha/2-Ep)$

#Generar Superficie 4

$s4=rg.\text{NurbsSurface}.\text{CreateFromCorners}(p13,p14,p16,p15)$

#Generar puntos patín inf en x=L

$p17=rg.\text{Point3d}(L,-Lp/2,-ha/2+Ep)$

$p18=rg.\text{Point3d}(L,-Lp/2,-ha/2)$

$p19=rg.\text{Point3d}(L,Lp/2,-ha/2)$

$p20=rg.\text{Point3d}(L,Lp/2,-ha/2+Ep)$

#Generar Superficie 5

$s5=rg.\text{NurbsSurface}.\text{CreateFromCorners}(p17,p18,p19,p20)$

#Generar puntos patin sup en x=L

$p21=rg.\text{Point3d}(L,-Lp/2,ha/2)$

$p22=rg.\text{Point3d}(L,-Lp/2,ha/2-Ep)$

$p23=rg.\text{Point3d}(L,Lp/2,ha/2-Ep)$

$p24=rg.\text{Point3d}(L,Lp/2,ha/2)$

$a=p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,p20,p21,p22,23,p24$

#Generar Superficie 6

$s6=rg.\text{NurbsSurface}.\text{CreateFromCorners}(p21,p22,p23,p24)$

#Generar Superficie 7

$s7=rg.\text{NurbsSurface}.\text{CreateFromCorners}(p9,p21,p22,p10)$

#Generar Superficie 8

s8=rg.NurbsSurface.**CreateFromCorners**(p1,p13,p15,p3)

#Generar Superficie 9

s9=rg.NurbsSurface.**CreateFromCorners**(p5,p17,p18,p6)

#Generar Superficie 10

s10=rg.NurbsSurface.**CreateFromCorners**(p1,p10,p22,p13)

#Generar Superficie 11

s11=rg.NurbsSurface.**CreateFromCorners**(p3,p5,p17,p15)

#Generar Superficie 12

s12=rg.NurbsSurface.**CreateFromCorners**(p12,p24,p23,p11)

#Generar Superficie 13

s13=rg.NurbsSurface.**CreateFromCorners**(p2,p14,p16,p4)

#Generar Superficie 14

s14=rg.NurbsSurface.**CreateFromCorners**(p8,p20,p19,p7)

#Generar Superficie 15

s15=rg.NurbsSurface.**CreateFromCorners**(p2,p11,p23,p14)

#Generar Superficie 16

s16=rg.NurbsSurface.**CreateFromCorners**(p4,p8,p20,p16)

#Generar Superficie 17

s17=rg.NurbsSurface.**CreateFromCorners**(p9,p12,p24,p21)

#Generar Superficie 18

s18=rg.NurbsSurface.**CreateFromCorners**(p18,p19,p7,p6)

s=[s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13,s14,s15,s16,s17,s18]

#Momento Actuante en Y kN*m

MyEd=(kN*1.35*pow((L/1000),2)/8)

#Deflec mm

d=(5*kN*pow(L,4))/(384*210000*Iy)

Anejo 6. Código en lenguaje de programación Visual Basic del componente “análisis”

```
Dim hw,tw,omega,fyw,bf,tf,fyf,epsilon,kj,chi,gamma,c_t,clase_alma,c_t_ala,
clase_ala As Double
    Dim bc, beff, be1,be2,bt,bn,area,tol,maxiter,iter,incremento,signo,res,yg As Double
    Dim em,zg,sigmasupext,sigmainfext,sigmasupint,sigmainfint,sigman1, sigman2,hplt,
hplc As Double
    Dim fcc,ftt As Double
    Dim hcc,htt,mom1,yg2,zg2,sigmasupext2,sigmainfext2,sigmainfint2,sigman3,fcc2,ftt2,
sigman4,hcc2,htt2,mom2 As Double
    Dim iz,iw,it,c1,ge,lcr,mcr,lambdalt1,alfalt1,filt1,chilt1 As Double
    Dim lambdalt2,alfalt2,filt2,chilt2,mltb1,mltb2 As Double
    Dim gm1,ar,katau,sigmae,taucr,lambdaw,chiw,vbwr,c As Double
    Dim med,vbfrd,vtot,vlim As Double
    Dim peso,mom1c1,sigmaintalac3,sigmaprom,altnplas,mom1c3 As Double
    Dim MyEd,I1,A1,A2,A3,I2,Iy,VEd As Double

hw = ha - (2 * Ep)
tw = Ea
omega = 4
fyw = ta
bf = Lp
tf = Ep
fyf = ta
epsilon = (235 / ta) ^ (1 / 2)
kj = 23.9
chi = -1
gamma = 1

'Cálculos Clase Alma
c_t = (hw - 2 * omega) / tw

If c_t <= 72 * epsilon Then
    clase_alma = 1
ElseIf c_t <= 83 * epsilon Then
    clase_alma = 2
ElseIf c_t <= 124 * epsilon Then
    clase_alma = 3
Else
    clase_alma = 4
End If
C_alma = clase_alma
```

'Cálculos Clase Ala

```
c_t_ala = (bf - 2 * omega) / 2 / tf
If c_t_ala <= 9 * epsilon Then
    clase_ala = 1
ElseIf c_t_ala <= 10 * epsilon Then
    clase_ala = 2
ElseIf c_t_ala <= 14 * epsilon Then
    clase_ala = 3
Else
    clase_ala = 4
End If
C_ala = clase_ala
```

'Cálculo lamda P y rho

```
Dim lambdap, rho As Double
lambdap = ((hw - 2 * omega) / tw) / (28.4 * epsilon * kj ^ 0.5)
If lambdap <= 0.673 Then
    rho = 1
Else
    rho = (lambdap - 0.055 * (3 + chi)) / lambdap ^ 2
End If
If rho > 1 Then
    rho = 1
End If
bc = hw / (1 - chi)
beff = rho * hw / (1 - chi)
be1 = 0.4 * beff
be2 = 0.6 * beff
bt = hw / (1 - chi)
bn = bc - be1 - be2
area = 2 * bf * tf + tw * (be1 + be2 + bt)
tol = 1
maxiter = 1000
iter = 0
incremento = 0.1
signo = 1
res = 10000000000
Do While math.Abs(res) > tol And iter < maxiter
    iter = iter + 1
    If iter = 1 Then
        yg = (bf * tf ^ 2 / 2 + bf * tf * (hw + 1.5 * tf) + hw * tw * (hw / 2 + tf) - bn * tw *
        (tf + bt + be2 + bn / 2)) / area
    Else
        If math.Sign(math.Sign(res) * signo) = 1 Then
            incremento = incremento
        Else
            incremento = incremento / 2
        End If
        If res > 0 Then
            yg = yg + incremento
        End If
    End If
    res = (yg - yg) / yg

```

```

Else
    yg = yg - incremento
End If
End If
em = bt + tf - yg
zg = hw + 2 * tf - yg
sigmasupext = fyf / gamma
sigmainfext = yg / zg * sigmasupext
sigmasupint = (zg - tf) / zg * sigmasupext
sigmainfint = (yg - tf) / yg * sigmainfext
sigman1 = (zg - tf - be1) / zg * sigmasupext
sigman2 = (zg - tf - be1 - bn) / zg * sigmasupext
hplt = yg * (sigmainfint - fyw) / sigmainfext
If sigman1 > fyw Then
    hplc = be1
Else
    hplc = zg * (sigmasupint - fyw) / sigmasupext
End If
fcc = bf * tf * (sigmasupext + sigmasupint) / 2 + 0.5 * (zg - tf) * sigmasupint *
tw - 0.5 * hplc * (sigmasupint - fyw) * tw - bn * tw * 0.5 * (sigman1 + sigman2)
ftt = bf * tf * (sigmainfext + sigmainfint) / 2 + 0.5 * (yg - tf) * sigmainfint * tw -
0.5 * hplt * (sigmainfint - fyw) * tw
signo = math.Sign(res)
res = fcc - ftt
Loop
hcc = (bf * tf * sigmasupint * (zg - tf / 2) + bf * tf * (sigmasupext - sigmasupint) / 2 *
(zg - tf / 3) + 1 / 2 * tw * (zg - tf) * sigmasupint * (zg - tf) * 2 / 3 - 1 / 2 * hplc * tw *
(sigmasupint - fyw) * (zg - tf - hplc / 3) - sigman2 * tw * bn * (zg - tf - be1 - bn / 2) -
1 / 2 * (sigman1 - sigman2) * tw * bn * (zg - tf - be1 - bn / 3)) / fcc
htt = (bf * tf * sigmainfint * (yg - tf / 2) + bf * tf * (sigmainfext - sigmainfint) / 2 *
(yg - tf / 3) + 1 / 2 * tw * (yg - tf) * sigmainfint * (yg - tf) * 2 / 3 - 1 / 2 * hplt * tw *
(sigmainfint - fyw) * (yg - tf - hplt / 3)) / ftt
mom1 = (hcc + htt) * fcc / 1000 / 1000
yg2 = (bf * tf ^ 2 / 2 + bf * tf * (hw + 1.5 * tf) + hw * tw * (hw / 2 + tf) - bn * tw *
(tf + bt + be2 + bn / 2)) / area
zg2 = hw + 2 * tf - yg2
sigmasupext2 = zg2 * fyw / (zg2 - tf)
sigmainfext2 = yg2 * sigmasupext2 / (zg2)
sigmainfint2 = (yg2 - tf) * sigmainfext2 / (yg2)
sigman3 = (bn + be2) * sigmasupext2 / (zg2)
sigman4 = (be2) * sigmasupext2 / (zg2)
fcc2 = bf * tf * (sigmasupext2 + fyw) / 2 + 0.5 * (zg2 - tf) * fyw * tw - bn * tw * 0.5 *
(sigman3 + sigman4)
ftt2 = bf * tf * (sigmainfext2 + sigmainfint2) / 2 + 0.5 * (yg2 - tf) * sigmainfint * tw
hcc2 = (bf * tf * fyw * (zg2 - tf / 2) + bf * tf * (sigmasupext2 - fyw) / 2 * (zg2 - tf / 3) +
1 / 2 * tw * (zg2 - tf) * fyw * (zg2 - tf) * 2 / 3 - sigman4 * tw * bn * (zg2 - tf - be1 -
bn / 2) - 1 / 2 * (sigman3 - sigman4) * tw * bn * (zg2 - tf - be1 - bn / 3)) / fcc2
htt2 = (bf * tf * sigmainfint2 * (yg2 - tf / 2) + bf * tf * (sigmainfext2 - sigmainfint2) /
2 * (yg2 - tf / 3) + 1 / 2 * tw * (yg2 - tf) * sigmainfint2 * (yg2 - tf) * 2 / 3) / ftt2
mom2 = fcc2 * (htt2 + hcc2) / 1000 / 1000

```

'lateral buckling

```
iz = tf * bf ^ 3 * 2 / 12
iw = iz * hw ^ 2 / 4
it = 1 / 3 * (2 * bf * tf ^ 3 + hw * tw ^ 3)
c1 = 1.132 'viene de kc=0.94
ge = 210000 / 2.6
lcr = L
mcr = (1 / 1000 / 1000) * (c1 * 3.1416 ^ 2 * 210000 * iz / lcr ^ 2) * (iw / iz + (lcr ^ 2 *
ge * it) / (3.1416 ^ 2 * 210000 * iz)) ^ 0.5
lambdalt1 = (mom1 / mcr) ^ 0.5
If hw / bf <= 2 Then
    alfalt1 = 0.49
Else
    alfalt1 = 0.76
End If
filt1 = 0.5 * (1 + alfalt1 * (lambdalt1 - 0.2) + lambdalt1 ^ 2)
chilt1 = 1 / (filt1 + (filt1 ^ 2 - lambdalt1 ^ 2)) ^ 0.5
lambdalt2 = (mom2 / mcr) ^ 0.5
If hw / bf <= 2 Then
    alfalt2 = 0.49
Else
    alfalt2 = 0.76
End If
filt2 = 0.5 * (1 + alfalt2 * (lambdalt2 - 0.2) + lambdalt2 ^ 2)
chilt2 = 1 / (filt2 + (filt2 ^ 2 - lambdalt2 ^ 2)) ^ 0.5
If chilt1 <= 1 Then
    mltb1 = chilt1 * mom1
Else
    mltb1 = 1 * mom1
End If
If chilt2 <= 1 Then
    mltb2 = chilt2 * mom2
Else
    mltb2 = 1 * mom2
End If
```

'cortante

'contribucion del alma

```
gm1 = 1
ar = hw * 4

katau = 5.34 + 4 * (1 / 3) ^ 2
sigmae = 189800 * (tw / hw) ^ 2
taucr = katau * sigmae
lambdaw = 0.76 * (fyw / taucr) ^ 0.5
chiw = 0.83 / lambdaw
vbwr = chiw * fyw * tw * hw / gm1 / 3 ^ 0.5 / 1000
```


'contribucion de las alas

```
c = ar * (0.25 + (1.6 * bf * tf ^ 2 * fyf) / (tw * hw ^ 2 * fyw))
med = 0.5 * (bf * tf * fyf * (hw + tf))
If c <= ar Then
    vbfrd = bf * tf ^ 2 * fyf / (c * gm1) * (1 - (med / (bf * tf * fyf * (hw + tf))) ^ 2) / 1000
Else
    vbfrd = bf * tf ^ 2 * fyf / (ar * gm1) * (1 - (med / (bf * tf * fyf * (hw + tf))) ^ 2) / 1000
End If
vtot = vbwrd + vbfrd
vlim = 1.2 * fyw * tw * hw / gm1 / 3 ^ 0.5 / 1000
```

'peso propio

```
peso = (2 * bf * tf + hw * tw) / 1000 / 1000 * 7850 * (L / 1000)
Peso_kg = peso
```

'posibilidad seccion clase 1-2

```
mom1c1 = (hw * hw * tw * fyw / 4 + fyf * tf * bf * (hw + tf)) / 1000 / 1000
```

'posibilidad seccion clase 3

```
sigmaintalac3 = hw / 2 * fyf / (hw / 2 + tf)
sigmaprom = (sigmaintalac3 + fyw) / 2
altnplas = fyw / fyf * (hw / 2 + tf)
mom1c3 = ((altnplas * tw * fyw / 2 * (4 / 3) * altnplas) + sigmaprom * tf * bf *
(hw + tf)) / 1000 / 1000
```

'Asignar clase

```
If clase_ala <= clase_alma Then
    Clase_Gen = clase_alma
Else
    Clase_Gen = clase_ala
End If
```

'Limitar Tamaño alma y alas

```
If (ha / bf) >= 1 And (ha / bf) <= 2 Then
    Peso_kg = peso
Else
    Peso_kg = 100000
End If
```

'Revisión viga

```
If Clase_gen = 1 Then
    Mc_Rd = mom1c1
ElseIf Clase_gen = 2 Then
    Mc_Rd = mom1c1
```

```

ElseIf Clase_gen = 3 Then
    Mc_Rd = mom1c3
Else
    Mc_Rd = mom1
End If
Vb_Rd = vtot

'Momento actuante
MyEd = (kN * math.Pow((L / 1000), 2) / 8) * 1.35

'Cortante Actuante
VEd = kN * 1.35 * (L / 1000) / 2

'Desplazamiento
'Cálculo de Intercia en Y
A1 = Ea * (ha - 2 * Ep)
A2 = Lp * Ep
A3 = Lp * Ep
I1 = (Ea * math.Pow(ha - 2 * Ep, 3) / 12) + A1 * math.Pow(0, 2)
I2 = (Lp * math.Pow(Ep, 3) / 12) + A2 * math.Pow(ha / 2 - Ep / 2, 2) * 2
Iy = I1 + I2
Desp = (5 * kN * math.Pow(L, 4)) / (384 * 210000 * Iy)
If Desp > (L / 300) Then
    Desp = 100000
Else
    Desp = peso
End If
Dim ratm, ratv, myed2, ved2, ratc, ratpl As Double

'Utilización
ratm = MyEd / Mc_Rd
ratv = VEd / Vb_Rd
ratpl = MyEd / mltb1
myed2 = ((kN * (L / 1000 / 2 / 2)) / 2) * ((L / 1000) - (L / 1000 / 2 / 2))
ved2 = kN * ((L / 1000 / 2) - (L / 1000 / 2 / 2))
ratc = (myed2 / Mc_Rd) + (ved2 / Vb_Rd)
If ratm > ratv And ratm > ratc And ratm > ratpl Then
    Utilizacion = ratm
ElseIf ratv > ratm And ratv > ratc And ratm > ratpl Then
    Utilizacion = ratv
ElseIf ratpl > ratm And ratpl > ratc And ratpl > ratv Then
    Utilizacion = ratpl
Else
    Utilizacion = ratc
End If
If Utilizacion > 1 Then
    Utilizacion = 100000
Else
    Utilizacion = peso
End If

```